An EA Process Template for PMs

Robert M. Flowe, Major, USAF
Defense Acquisition University

This presentation represents the views of the author, and does not represent the views of the Department of Defense, the Air Force, or Defense Acquisition University

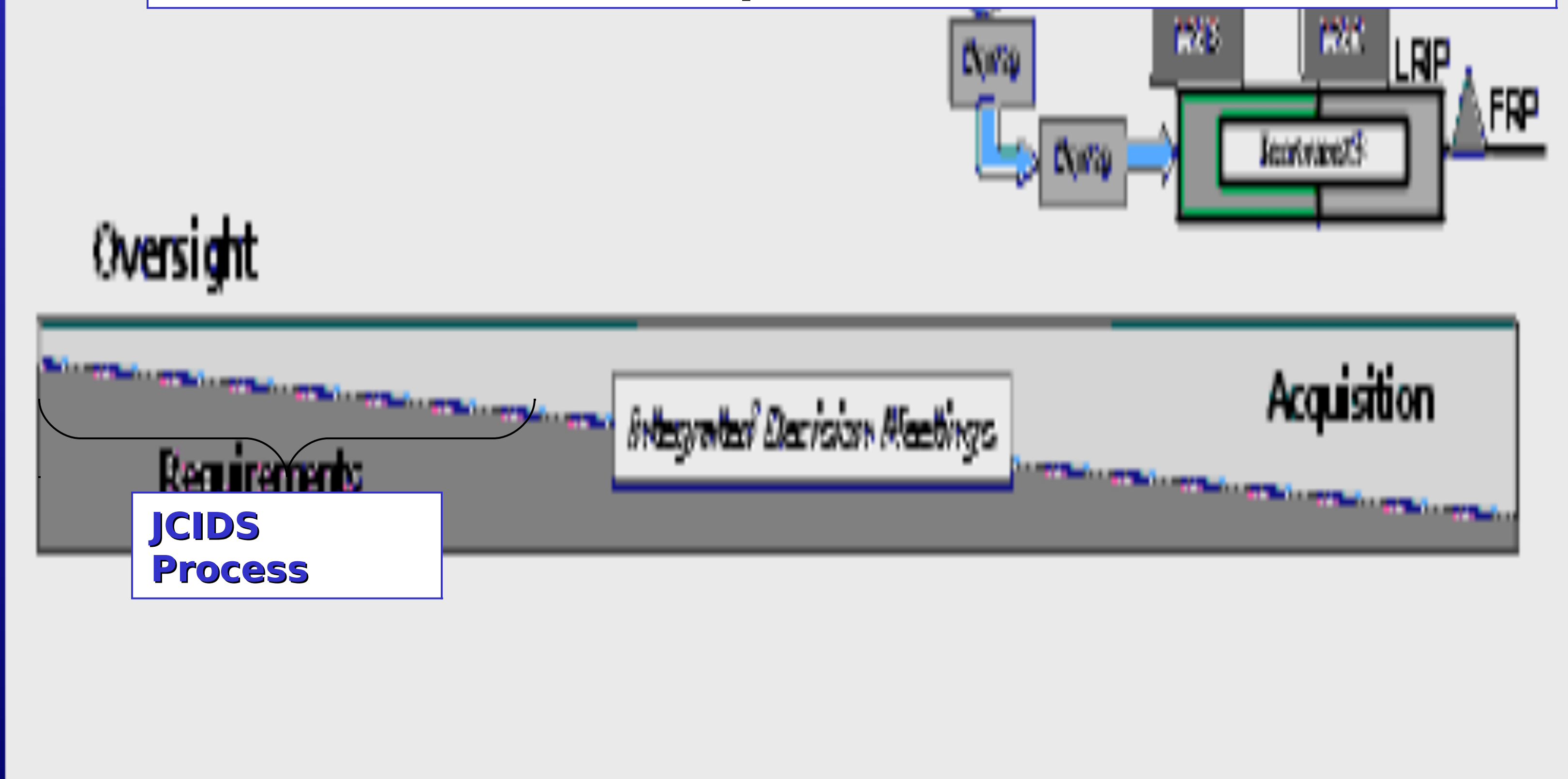
Agenda

- The “New” Acquisition Environment
- Definitions: Evolutionary Acquisition & Spiral Development
- Why an EA Process Template?
- Key Features of this Template
- Process Description
- Special Interest Items:
  - Contracting Implications
  - Logistics Implications
  - Test and Evaluation Implications

The New Acquisition Environment

Defense Acquisition University

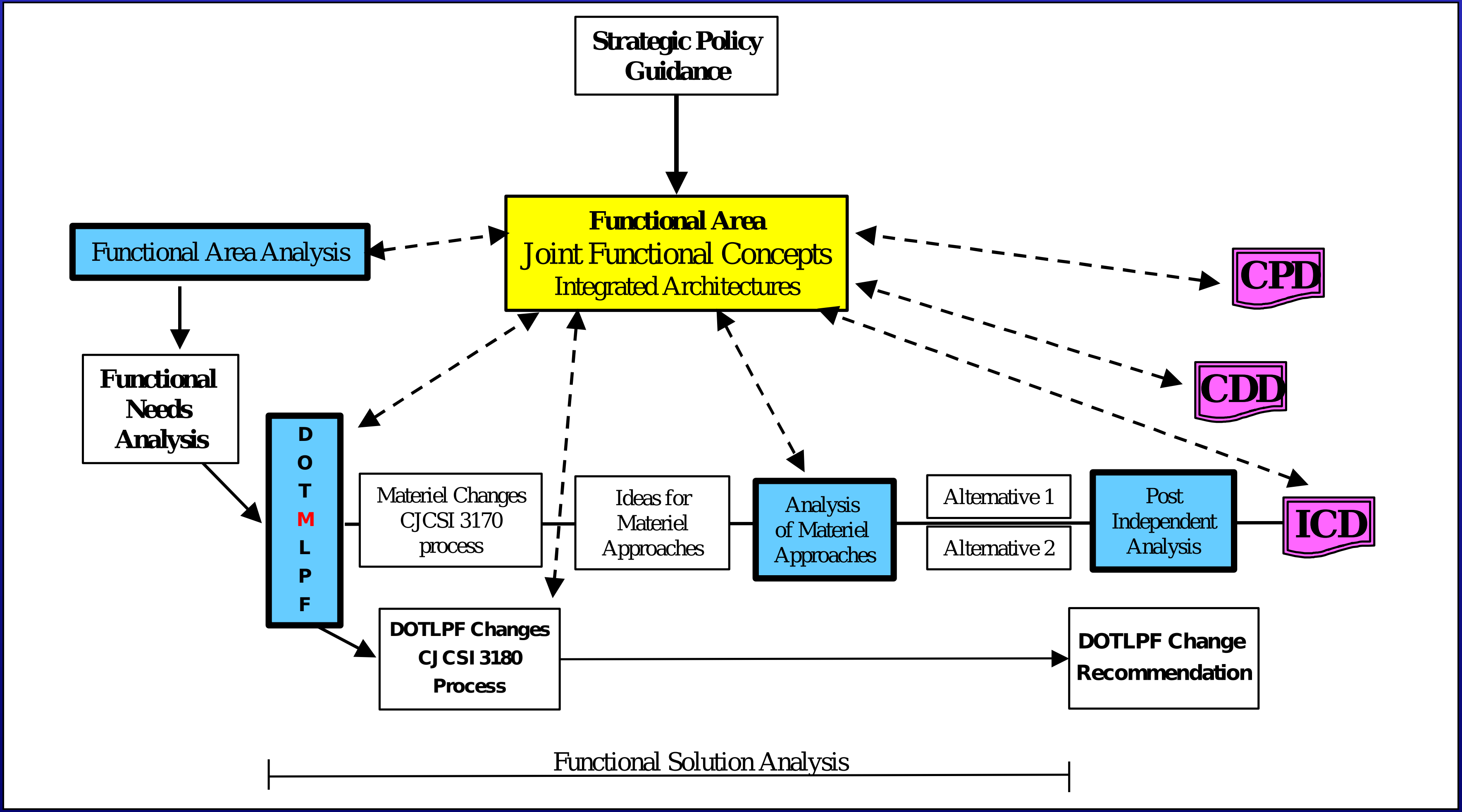
Joint Capabilities Integration & Development System
And Acquisition Process



Source: DRAFT Attachment 2 to SecDef Memo, Ops of the Defense Acquisition System, September 18

JCIDS Process

Defense Acquisition University



Source: CJCSM 3170.01M FLAG STAFFING DRAFT April 2003

Evolutionary Acquisition

Evolutionary acquisition is DoD's preferred strategy for rapid acquisition of mature technology for the user. An evolutionary approach delivers capability in increments, recognizing, up front, the need for future capability improvements. The success of the strategy depends on the consistent and continuous definition of requirements and the maturation of technologies that lead to disciplined development and production of systems that provide increasing capability towards a materiel concept.

Source: **DRAFT** Attachment 2 to SECDEF memo, *Operation of the Acquisition System*, dated 18 September 2002

Spiral Development

A process in which a "…desired capability is identified, but the end-state requirements are not known at program initiation. Those requirements are refined through demonstration and risk management; there is continuous user feedback; and each increment provides the user the best possible capability. The requirements for future increments depend on feedback from users and technology maturation."

Source: DRAFT Attachment 2 to SecDef Memo, Ops of the Defense Acquisition System, September 18, 2002

When to Apply EA

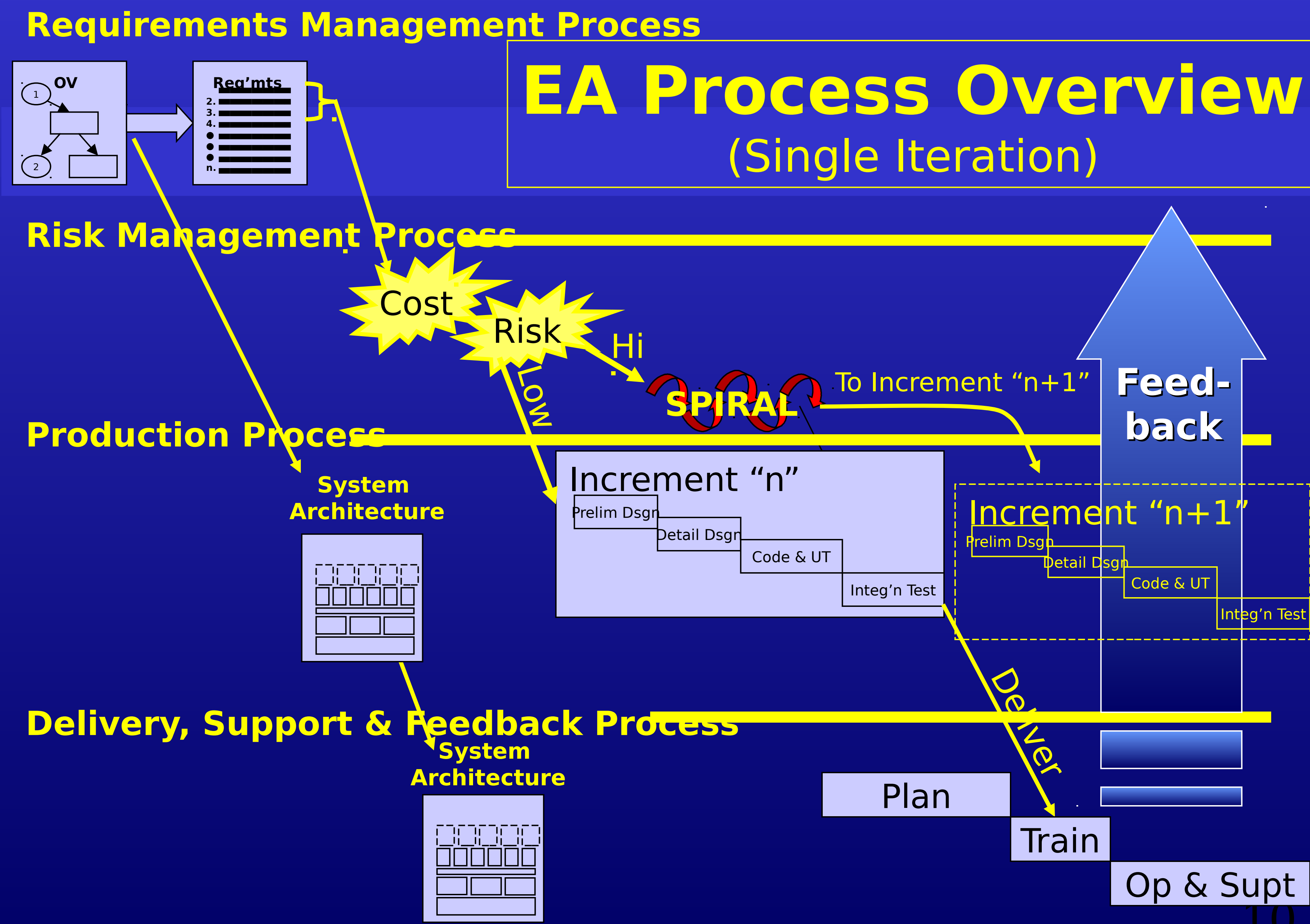
- Requirements volatility
  - Changing requirements
  - Functional requirements
- Rapidly-evolving technology
- Rapidly-evolving threat
- Resource volatility

Why an EA Process Template?

- Most EA literature and policy discusses EA in general terms
- PMs are left to figure out how to implement EA for their programs
- Not much “how to” or “why” information available
- This notional EA process template describes key features and functions that must exist, in some form, for EA to work
- This is not a cookbook “one size fits all” solution

Key Features of This EA Approach

- Acquirer & User collaborate in requirements definition & domain analysis to develop an <u>integrated architecture</u>
- Operational, System, and Technical Views used to identify, validate, prioritize, and order requirements
- Continuous management of cost and development risk
- Frequent feedback between acquirer and <u>all stakeholders</u> throughout development & support
- Four major concurrent processes:
  - Requirements Management
  - Risk Management
  - Production (including test & evaluation)
  - Delivery, Support & Feedback


EA Process Overview
(Single Iteration)
Requirements Management Process
OV
1
2
Req'mts
2.
3.
4.
n.
Risk Management Process
Cost
Risk
Hi
Low
SPIRAL
To Increment "n+1"
Feed-back
Production Process
System Architecture
Increment "n"
Prelim Dsgn
Detail Dsgn
Code & UT
Integ'n Test
Increment "n+1"
Prelim Dsgn
Detail Dsgn
Code & UT
Integ'n Test
Deliver
Delivery, Support & Feedback Process
System Architecture
Plan
Train
Op & Supt

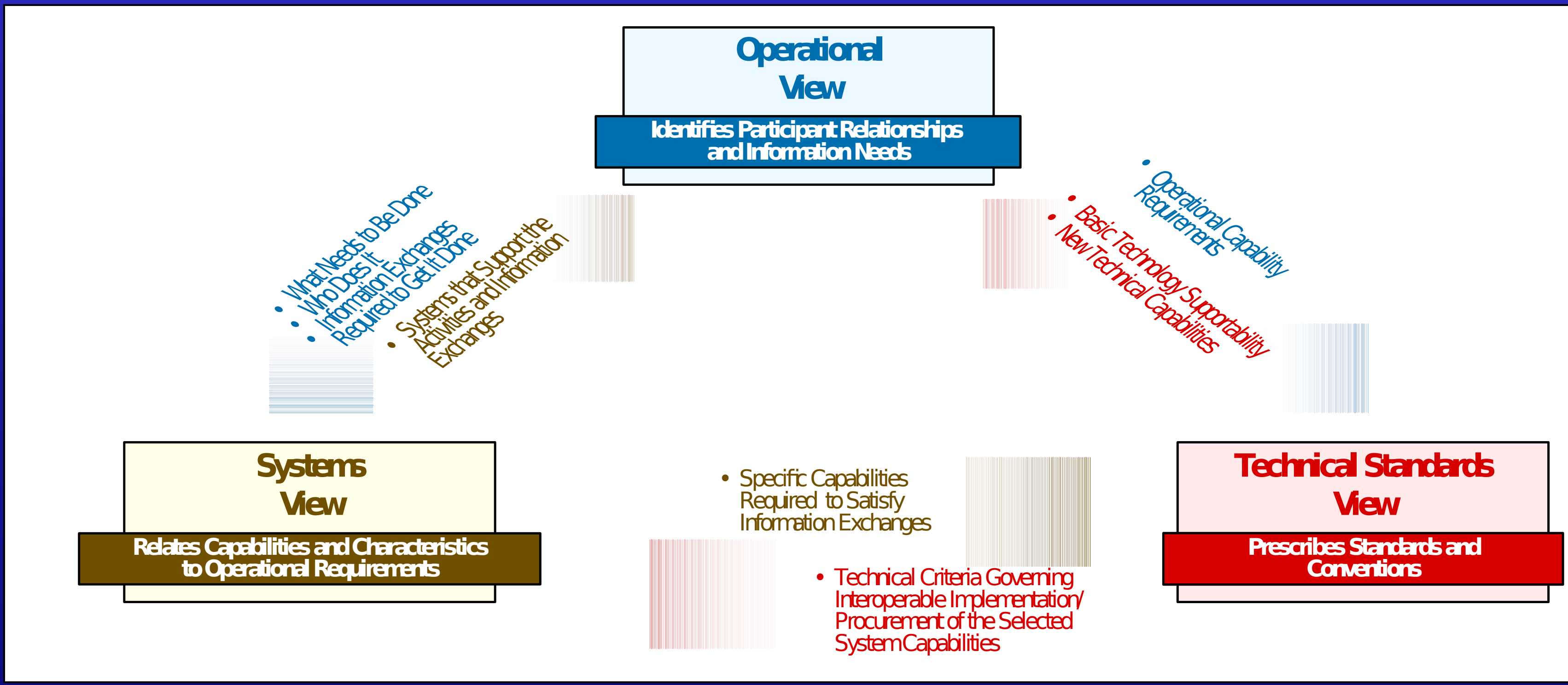
Investment in Architecture

Requirements and Architecture

- Requirements don’t “just happen”
- Requirements reflect functions to be performed to provide capabilities that fulfill operational needs
- Requirements must therefore be mapped to operational needs (this is not new)
- For the acquirer to understand requirements, the acquirer must first understand the User’s operational domain
- The mechanism for this understanding is the integrated architecture and its related views

Integrated Architecture

Defense Acquisition University



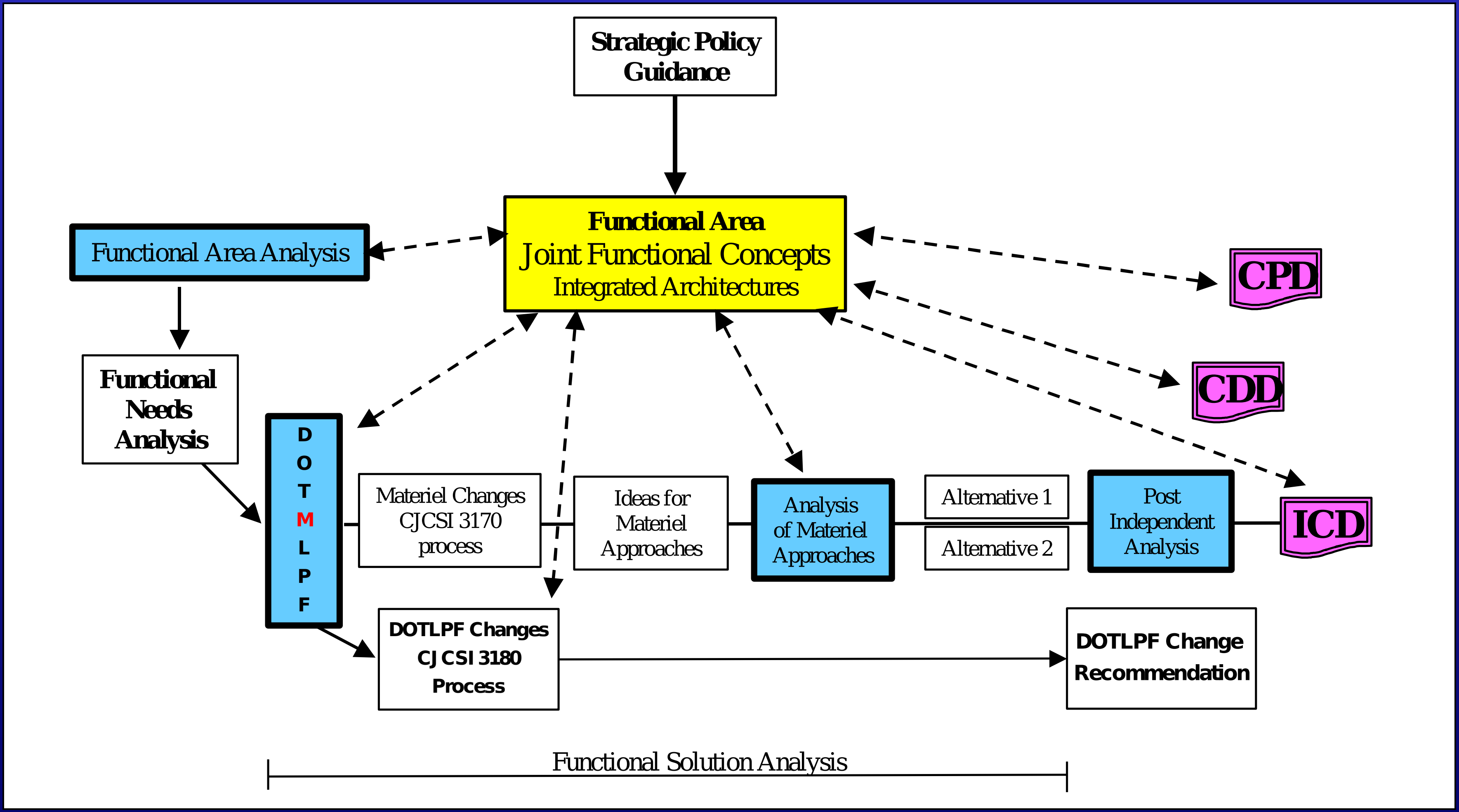
Architecture: the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time.

DoD Integrated Architecture Panel, 1995, based on IEEE STD 610.12

Source: DoD Architecture Framework, Version 1.0, Volume 1 [Final Draft], January 2003

Architectures: JCIDS Input

Defense Acquisition University



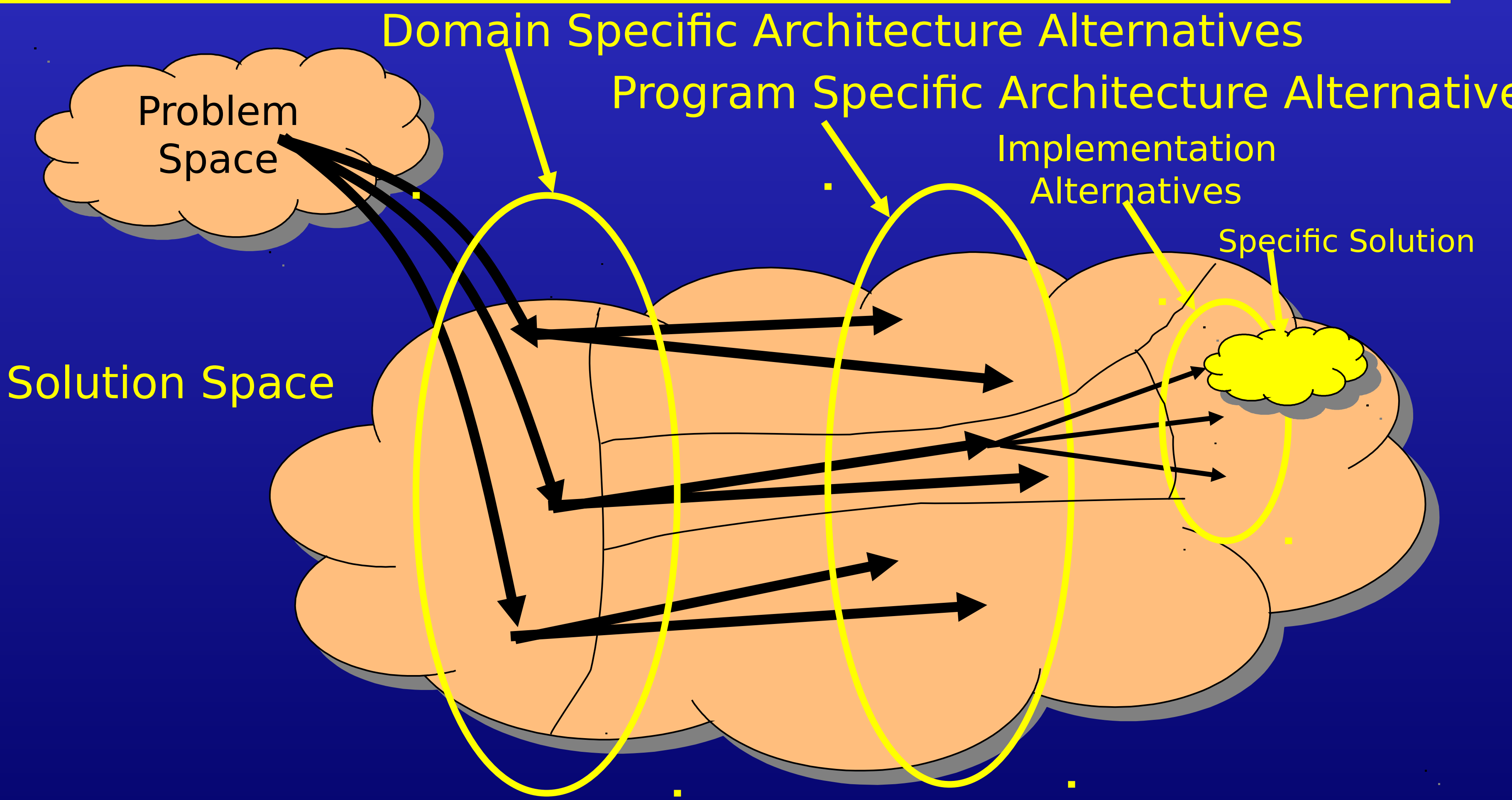
Source: CJCSM 3170.01M FLAG STAFFING DRAFT April 2003

Program Level Architecture

- Architectures exist at many levels
- Enterprise-wide architectures address the “Big Picture”
  - Joint Operational Architecture
  - Joint Technical Architecture
- These are insufficiently detailed to inform the development of program-level requirements
- A program-level integrated architecture must be developed for program-specific requirements
- The Program-level architecture must be consistent with higher-level architectures
- This investment in program-level architecture must occur before Milestone B (the earlier the better!)

Domain and Program-Specific Architectures

Defense Acquisition University

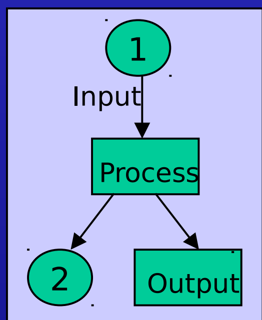


Based upon USC-CSE DSSA course, January 2001

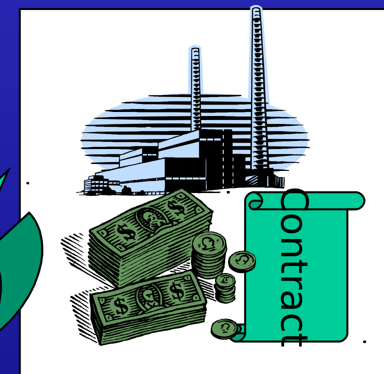
Views of the Solution Space

Defense Acquisition University

Operational View

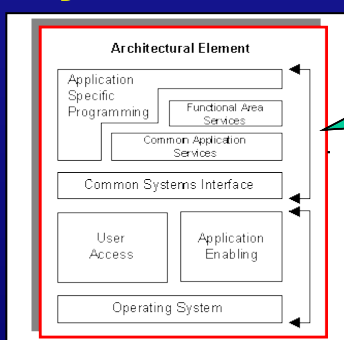


Business View



Solution

Systems View



Technical View

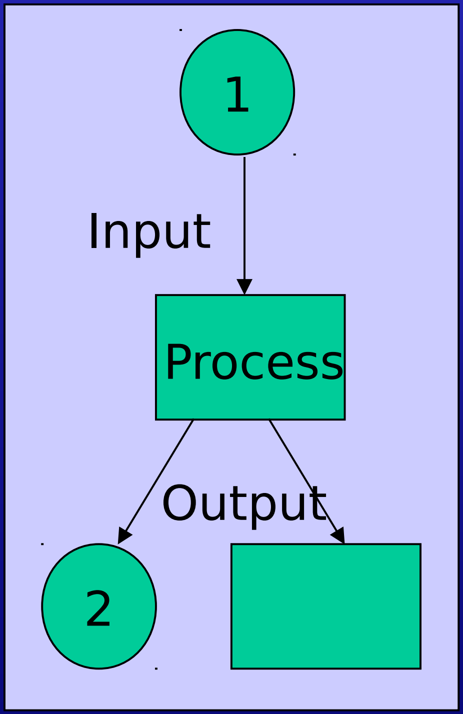
Also consider:

- Logistics view
- Security view
- Other views

Operational View

Defense Acquisition University

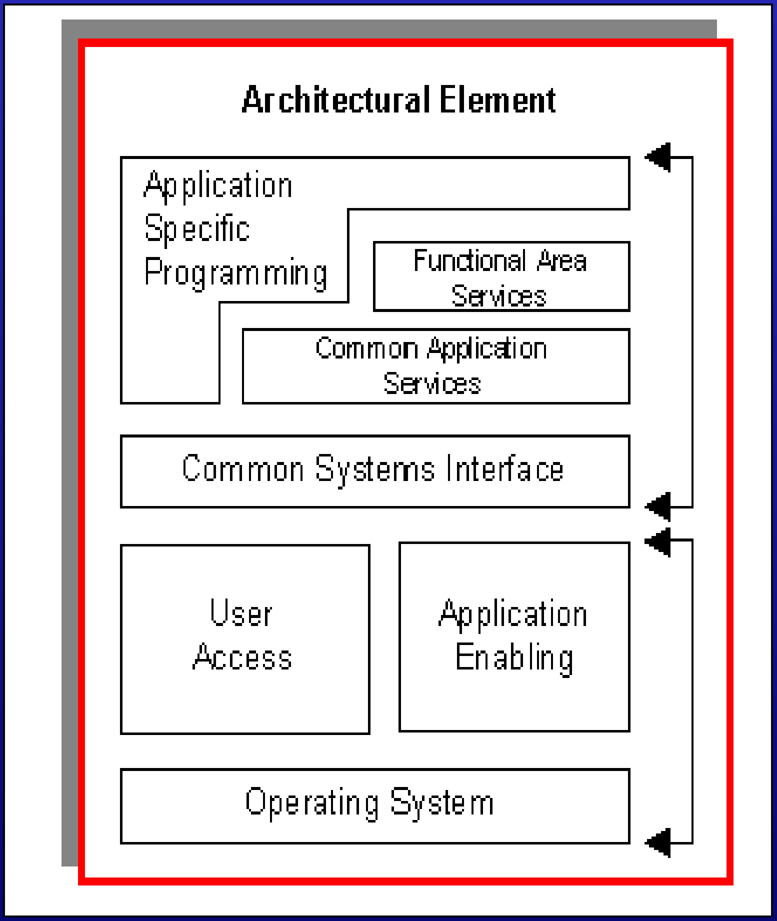
OV



- Represents how the User operates
  - Current and future doctrine, tactics, techniques, & procedures.
  - Time-phased: describes required capability growth over time
- Focused on User, facilitated by Acquisition, S&T Communities
- Coupled with domain modeling to allow first-order tradeoffs on capability vs. cost vs. time
  - Balances "What is Needed" with "What is Possible"
- OV must be validated by User, endorsed by acquirer
  - Both must "own" the OV and related System View (SV)

The OV is a key instrument for achieving and maintaining acquirer/user/stakeholder understanding

Systems View
(sometimes called "Systems Architecture")
Defense Acquisition University

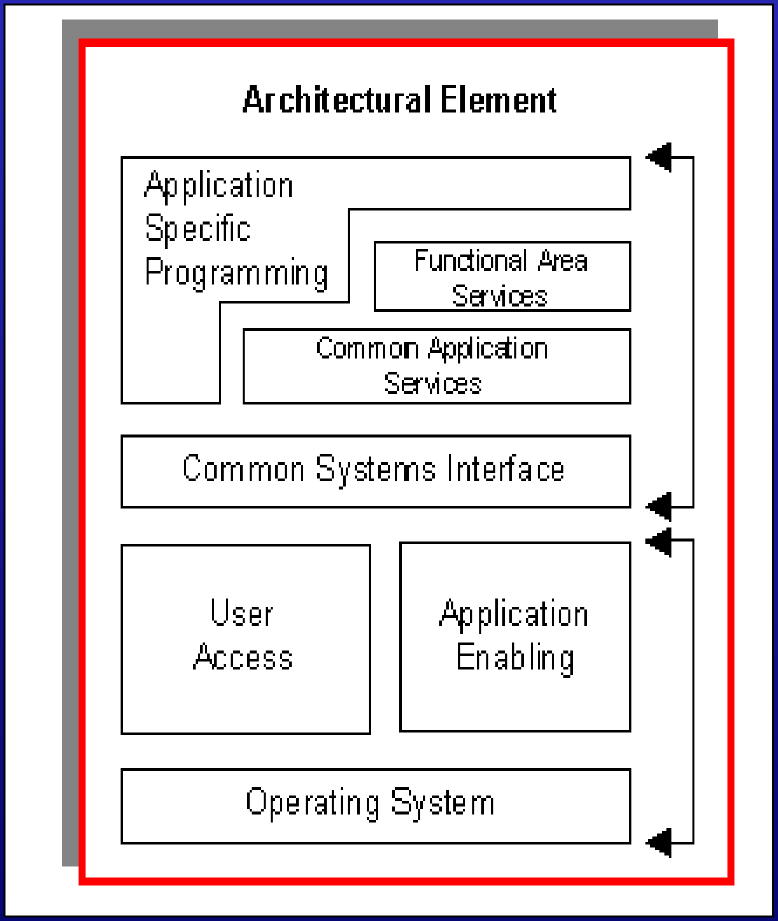


- SV strongly influenced by the OV
  - Output of domain analysis & modeling
- SV describes functional "boundaries" of the program, interaction rules, and key interfaces
- Provides framework within which functional capability will operate and evolve
- SV reflects goals of flexibility, scalability, interoperability, robustness, security, etc.
- Establish or adopt an <u>enterprise</u> SV
  - Standard architectures exist for some domains
  - These enhance interoperability, facilitate reuse, and reduce risk
  - Standardize interfaces when possible

Graphic: Barry M. Horowitz, Ph.D.ESC-TR-94-208, September 1994

Systems View

Defense Acquisition University



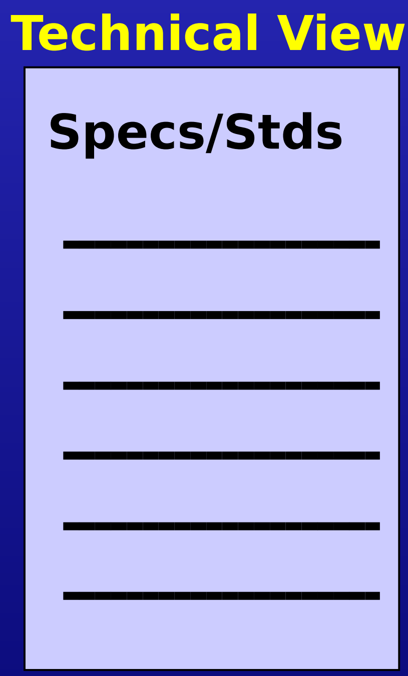
- SV also influenced by requirements allocation
  - Grouping of dependent functions define functional modules
  - Required interaction between functional modules defines interface requirements
  - Aggregated functional interactions drives system-level design
    - Capacity, throughput, bandwidth, scale, processor speed, network topology, etc.

Graphic: Barry M. Horowitz, Ph.D.ESC-TR-94-208, September 1994

Systems Architecture as a Deliverable

- Modular, scalable, open systems architecture minimize integration and support cost over life cycle
  - Core functionality & services allocated to the architecture
  - Mission-specific functionality allocated to modular elements
  - Functional modules can be added and changed without extensive impact to architecture or other elements.
  - Support for core will be similar across configurations
  - Unique support requirements limited to functional modules
  - Retrofit/upgrade cost is minimized
- Keep architecture distinct from functionality
  - System architect maintains architecture and imposes enterprise rules on functional elements
  - Maintaining “separation of powers” reduces likelihood of integrator monopoly
    - Consider making interface specification public

Technical View

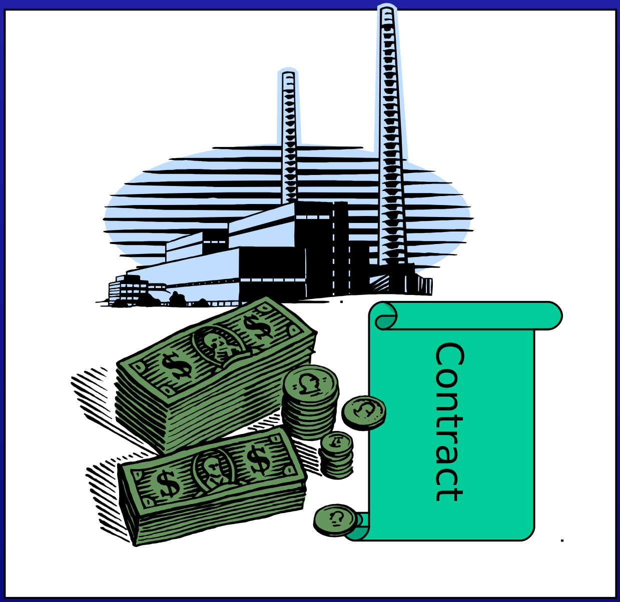


- Technical view of the integrated program-level architecture
- Describes the specifications and standards that apply to the domain for which the solution is an instance
  - Tailored as appropriate, with an eye toward maintaining interoperability
- Adherence to technical architectural standards reduces the engineering tradespace
  - promotes reuse
  - enhances (but does not guarantee) interoperability

Business View

Defense Acquisition University

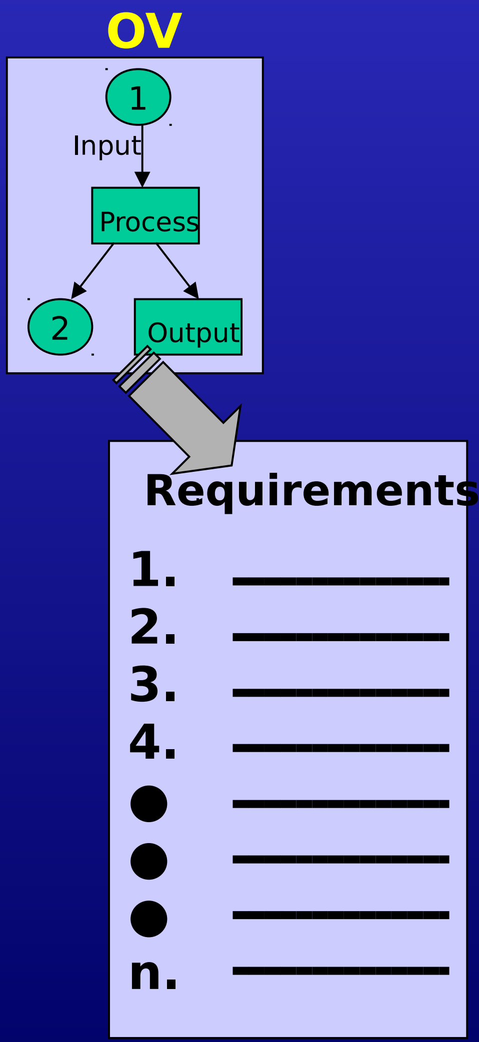
Business View



- Business View (for lack of a better term) provides insight into the organizational requirements and constraints of the enterprise
- Illuminates
  - Acquisition strategy
  - Contract structures
  - Resource streams
    - Money
    - Personnel
  - Political issues
- Highlights the integration and interoperability requirements of these facts

Deriving Requirements from Architectures

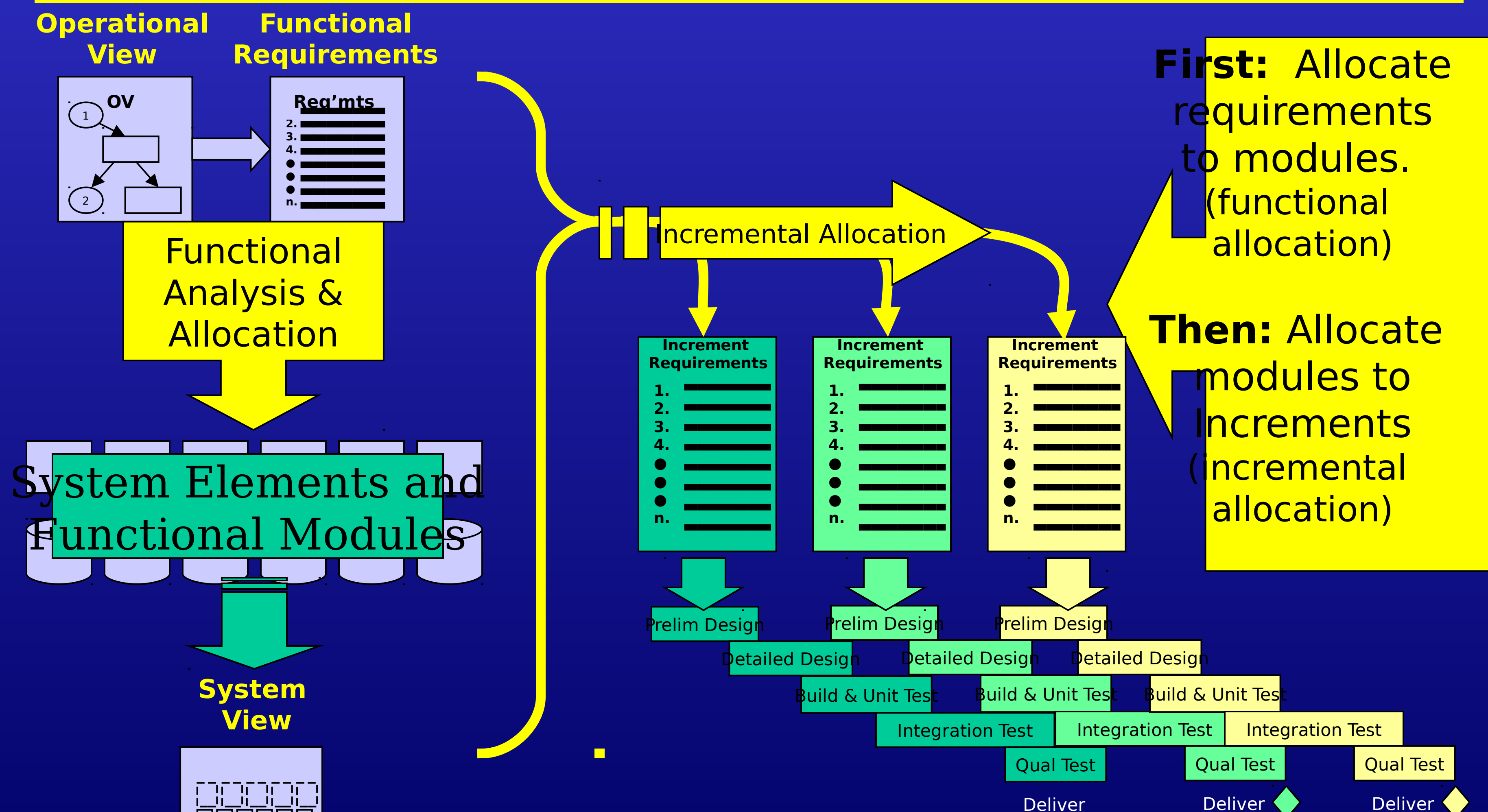
Defense Acquisition University



- OV describes organizational processes, inputs and outputs
  - This suggests functional capabilities required
    - What the system must do
    - Prioritized in order of operational relevance & timing
- Operational scenario analysis is useful
  - Users “walk through” their missions
  - Helps identify priorities, context, conditions, requirements, and functional dependencies
- <u>Important</u>: Dependencies exist between requirements
  - Group requirements based upon functional coherence and dependence
  - Allocate similar/related requirements to functional modules—“functional allocation”
- Functional allocation informs the development of the systems view of the integrated architecture
  - Distinguish functional elements from system services
- Priority of functional capabilities informs the time-phasing of requirements

Functional and Incremental Allocation

Defense Acquisition University

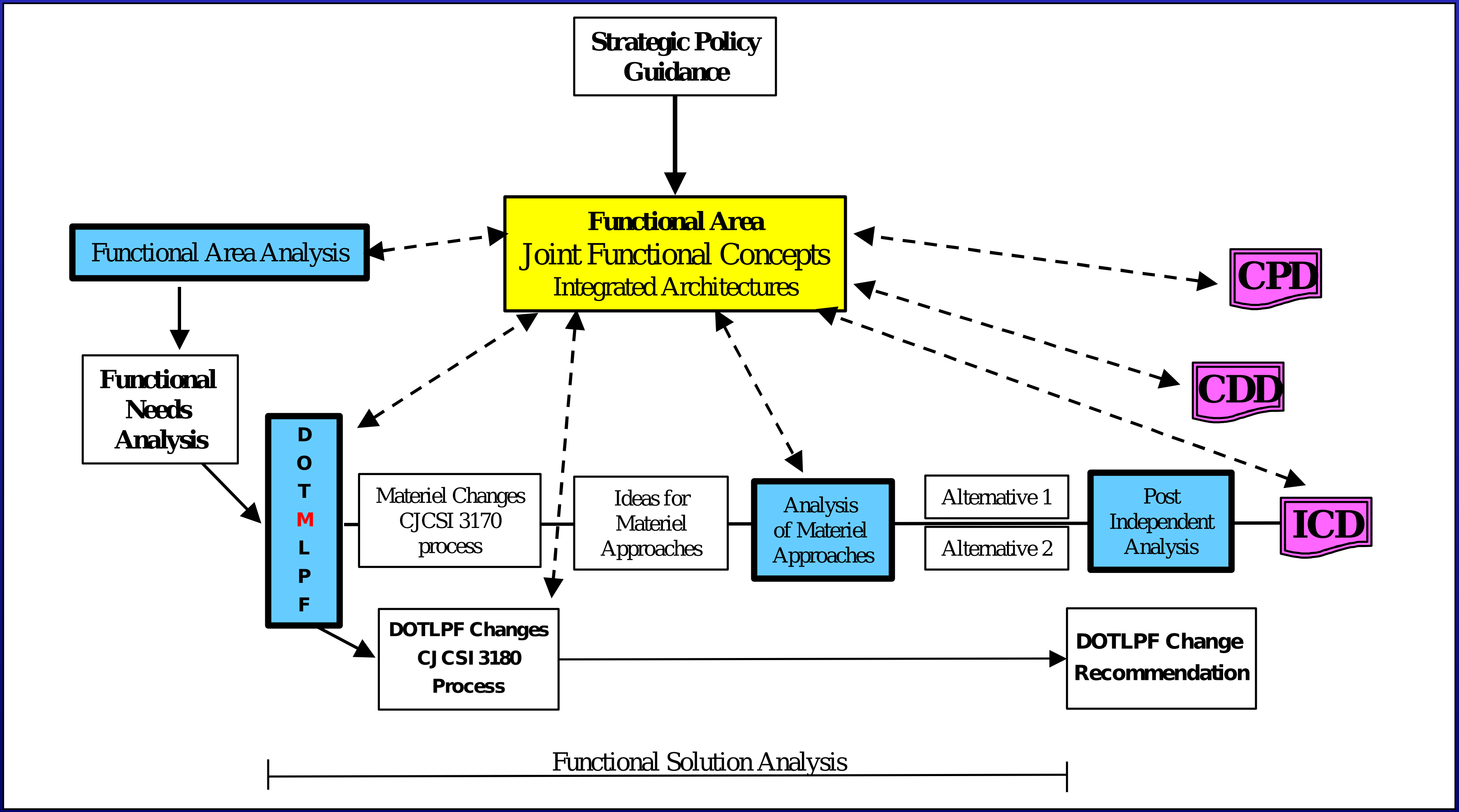


Functional Allocation must minimize coupling between modules
Incremental Allocation must minimize coupling between increments

Process: Requirements Management

Requirements: JCIDS Input

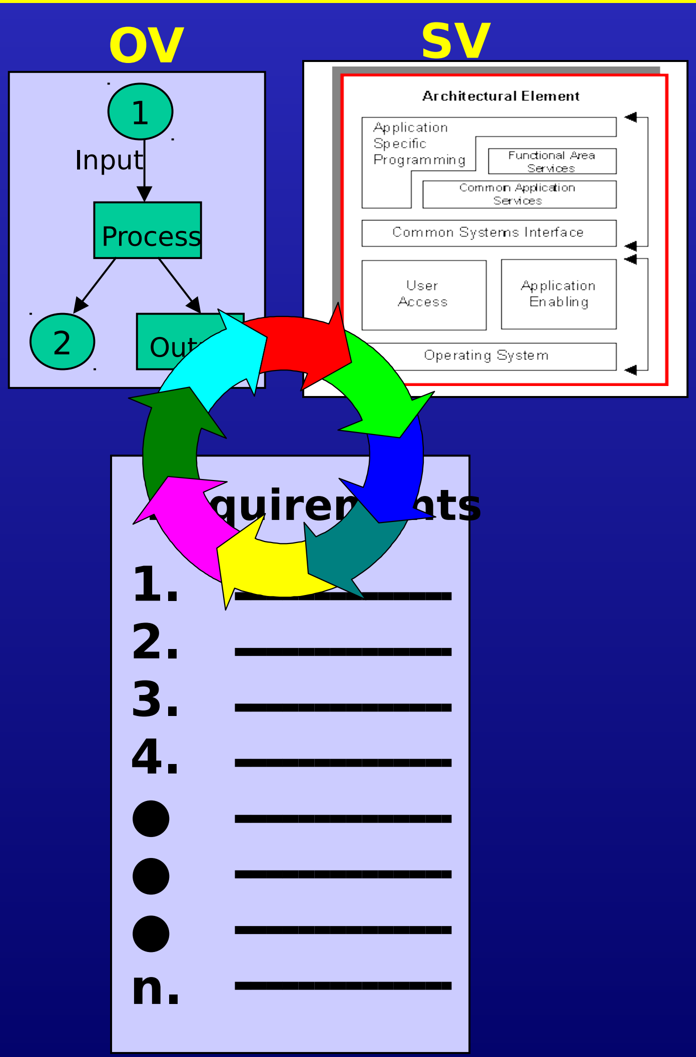
Defense Acquisition University



Source: CJCSM 3170.01M FLAG STAFFING DRAFT April 2003

OV, SV and Requirements

Defense Acquisition University



- OV drives functional requirements
  - Linkage between OV and functional requirements must be maintained as requirements are allocated to modules and increments.
- SV supports OV
  - Scope and scale of system required by OV must be supported by the systems architecture
- SV bounds functional requirements
  - Defines the manner in which requirements must be implemented
  - Establishes "enterprise rules" to govern interaction among elements
- This is an iterative and dynamic process that requires active and persistent management

Requirements Management

- Once OV and SV are established, requirements are (somewhat) bounded
  - OV describes what must be done, how, with whom
  - SV defines how the system is implemented
  - Requirements may be elaborated and re-prioritized within these constraints
- EA recognizes that operational needs can change
  - But these changes need to be validated against and reflected in the OV

Requirements Management

- PMO & Architect must
  - Periodically reconvene the stakeholders to formally reassess the stated requirements, their functional and incremental allocation, and implementation priority
  - Use the OV and SV to communicate the boundaries within which the program is defined
  - Incorporate results from field use, experimentation, test and evaluation
  - Facilitate discussion, capture the evolving understanding of the problem and solution spaces
    - Demonstrate prototypes and work-in-progress to make future deliverables tangible
    - Use modeling and simulation to illustrate concepts, to elicit and clarify requirements
- This is not easy, nor is it inexpensive!

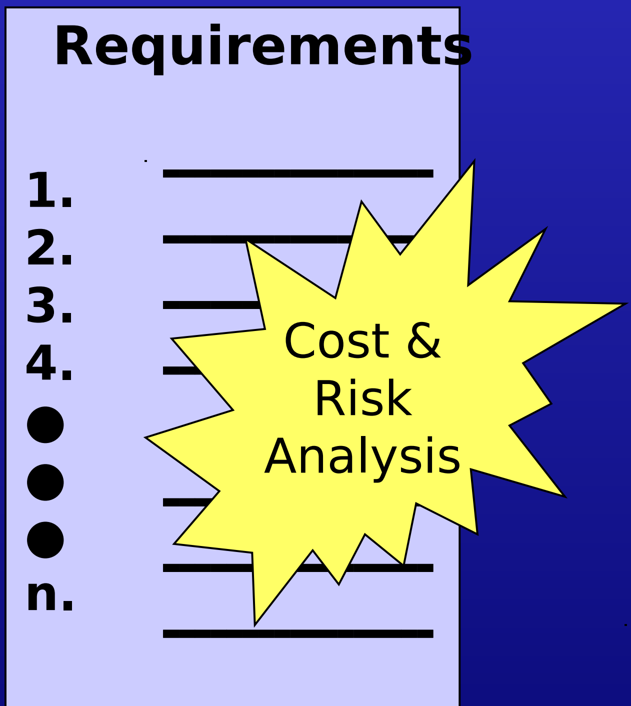
Requirements Management

- Every requirement arises from, and thus must be mapped to the OV
- Any “new” requirement is either an elaboration of an existing requirement, or results from a change to the OV
- In the “contract” with the User, elaborated requirements should be considered “in scope”
  - A natural result of gaining a clearer understanding of the actual nature of the requirement
  - Certainly has cost implications, but needs to be handled within the program constraints
- Once underway, major changes to the OV require readjustment to program resources and plans
  - Mission/doctrine changes that could not be anticipated
  - Major changes that require reexamining resources assumptions
  - This discussion must be held in concert with ALL stakeholders!

Process:
Risk Management

Assess Cost & Risk

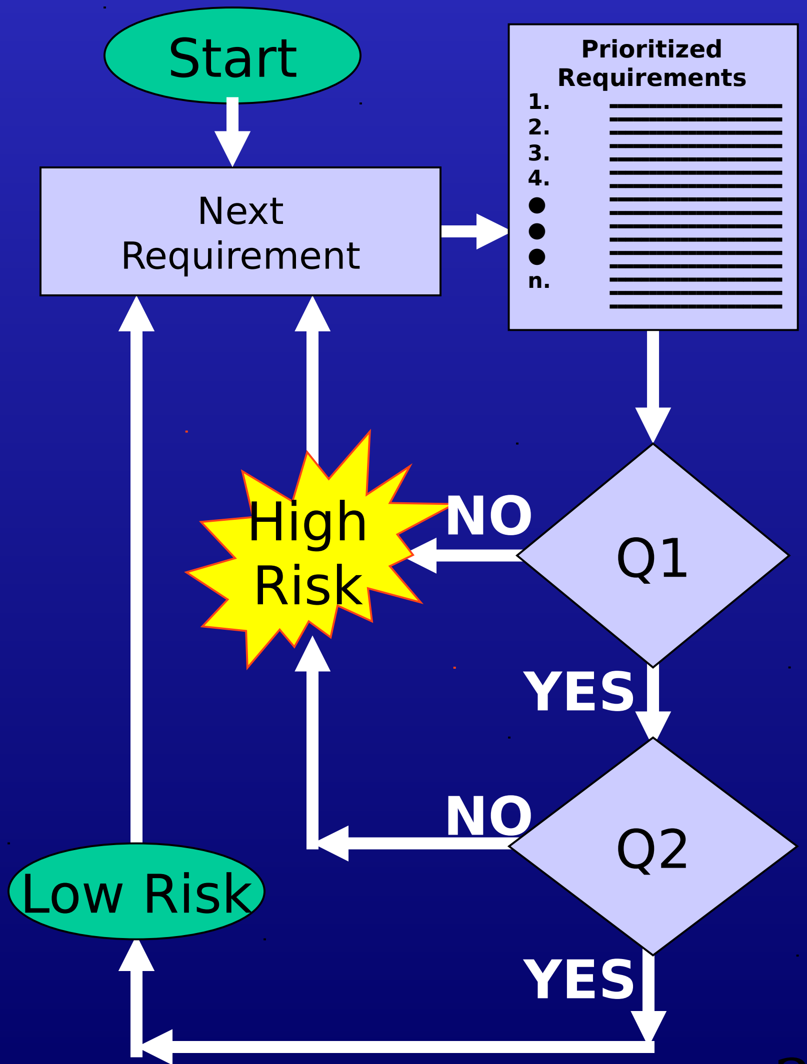
Defense Acquisition University



- Conduct cost and risk analysis
  - of each requirement
  - to reflect current understanding and technical maturity
  - to identify further dependencies
- Recognize schedule and budget constraints
  - Each requirement represents a “claim” on resources: budget & schedule
  - PM must ensure commitments don’t exceed resources for any given increment
- Provide input to budget & resource planning

Requirements Risk Assessment

- For <u>EACH REQUIREMENT</u>
- You must answer 2 fundamental questions:
  1. Do I understand what is required?
  2. Does a solution exist?
- Answer “NO” or “I’m not sure” to Q1 or Q2 means HIGH RISK (until proven otherwise)
- High Risk requirements demand special attention before implementation (more on this later)

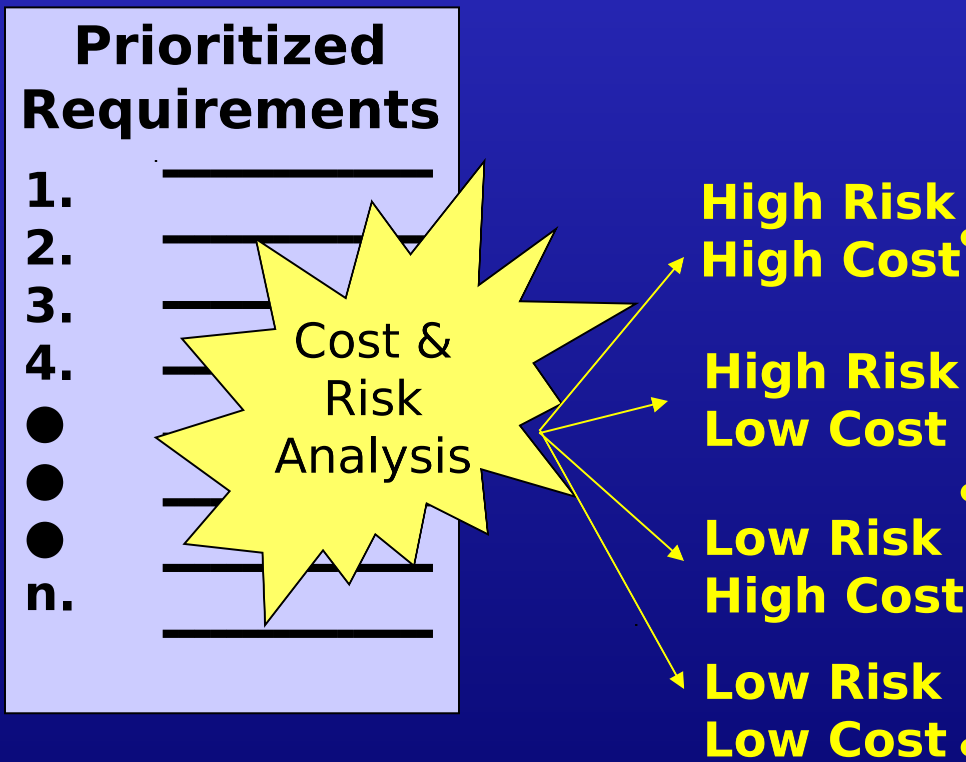


Requirements Cost & Risk

Defense Acquisition University

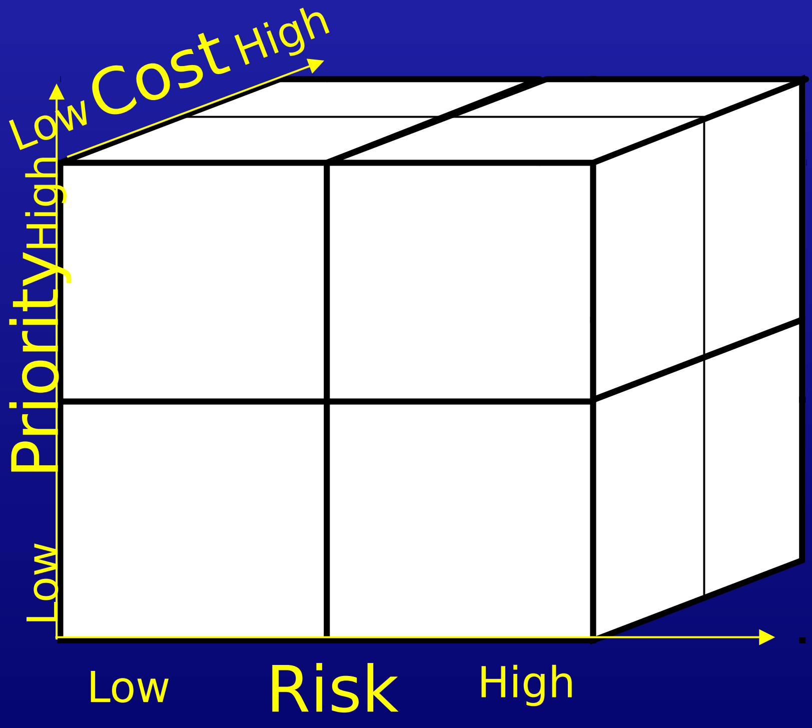
- Apply standard cost estimating methodologies to requirements deemed “Low Risk”
- “High Risk” requirements require further scrutiny
  - What is the nature of the risk?
    - Poorly-articulated or understood requirement statement?
    - Current technology incapable of meeting specified performance?
  - What is the “magnitude” of the risk?
    - Do the Users know what they want?
    - How big is the technology gap?
- Risk mitigation scope and cost will depend upon this analysis
- Risk increases the expected cost and variance for any requirement

Assess Cost & Risk (continued)



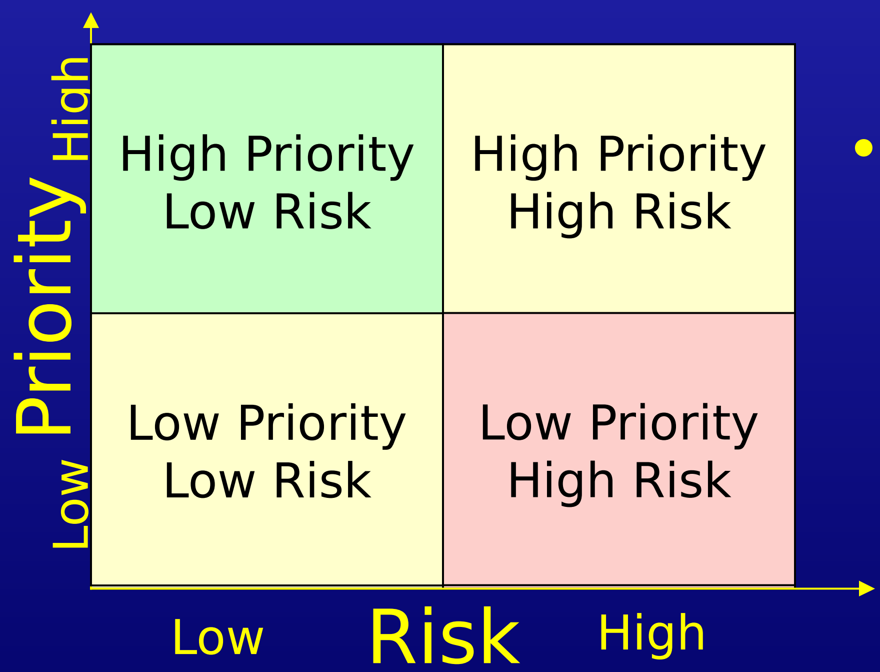
- Users, through OV and domain modeling, establish the priority for each requirement
- PMO analysts establish initial risk and cost attributes for each requirement
- These priority, cost, and risk assessments are preliminary, and are used for planning
- Continuous, proactive risk/cost management ensures affordability of build plan

Assess Cost & Risk (continued)



- Each requirement will have attributes of priority, cost, and risk
- Attributes can be expressed in a matrix
- Note that risk and cost are generally correlated

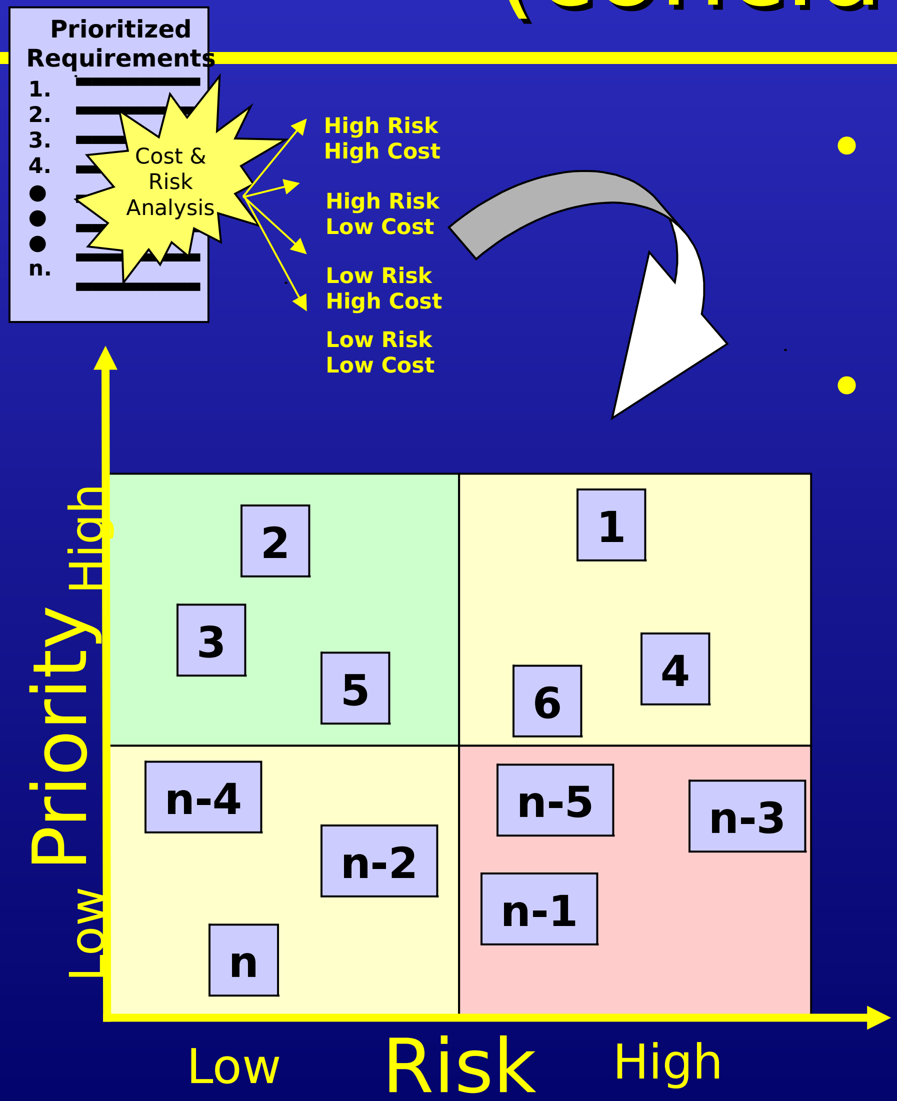
Assess Cost & Risk (continued)



- Since cost and risk tend to be correlated, the 3D matrix can be simplified into a 2D priority vs. risk matrix
- Risk/Priority matrix provides a guide to
  - implementation sequence
  - risk management strategies
  - resource allocation priorities

Assess Cost & Risk (concluded)

Defense Acquisition University



- Each requirement is mapped to the risk/priority matrix
- Use this framework to
  - Allocate requirements to increments
  - Develop risk management strategies
  - Facilitate resource planning
  - Conduct tradeoffs with Users & other stakeholders

Mitigate Risks

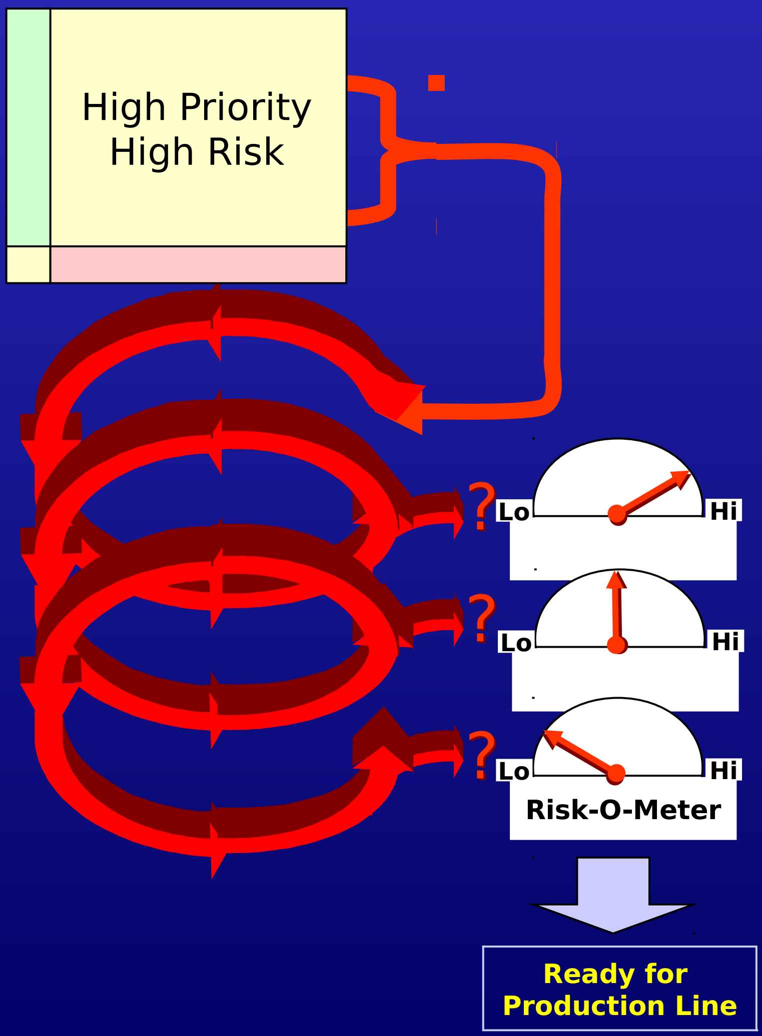
Defense Acquisition University

- For **High-Priority/High-Risk** requirements
  - Develop mitigation strategy based upon nature, magnitude of risk (reference previous cost/risk analyses)
  - Do not defer simply because risk is high;

-BUT-

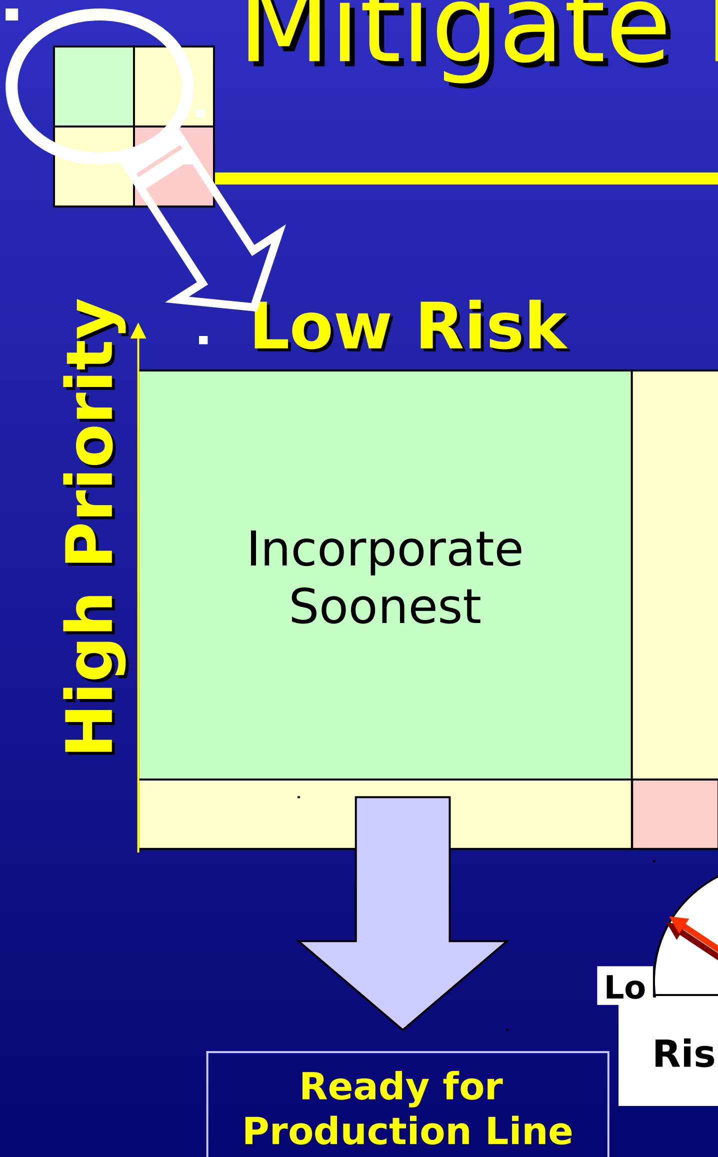
  - <u>Do not implement until risk is reduced</u>

Mitigate Risks (continued)

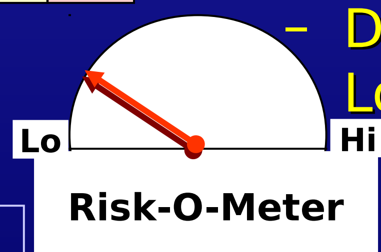


- Risk mitigation approach depends upon nature of risk
- Common sources of Risk
  - poorly-specified requirements
  - technical immaturity
- Spiral Development a useful tool
  - Rapid prototyping with User to resolve requirements ambiguities
  - Iterative prototypes to resolve technical maturity issues
- Keep prototyping activities focused
  - Establish limits on cost and duration
  - Tailor output to feed production line
- Exit Criterion: Low Implementation Risk

Mitigate Risks (continued)

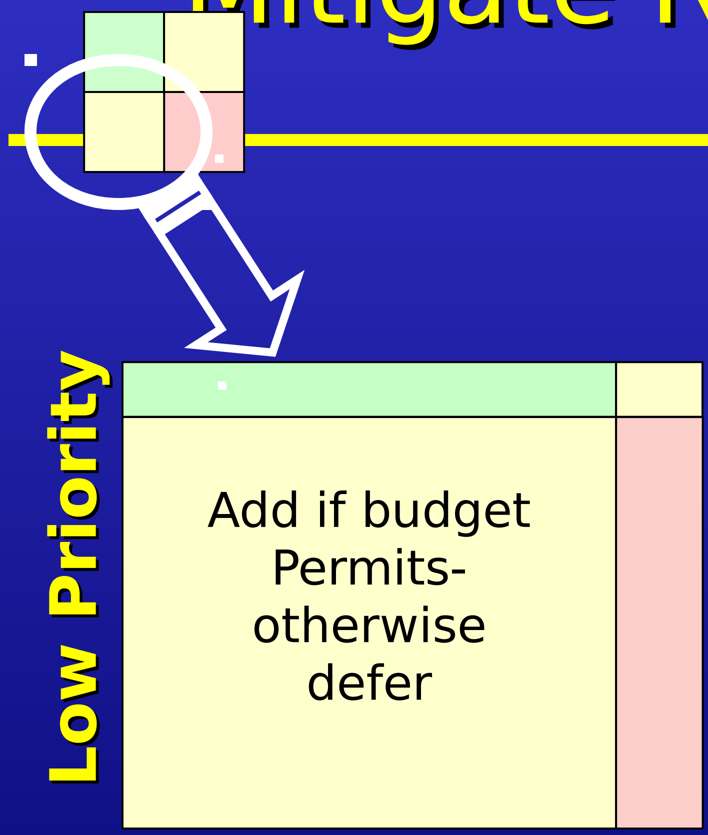


- For **High-Priority/Low-Risk** requirements
  - Allocate to increment based on
    - Priority
    - Resource and other constraints
    - Dependencies with other requirements
  - Don't devote entire budget to Low Risk requirements
    - High Risk "Gotchas" are out there waiting to bite you!

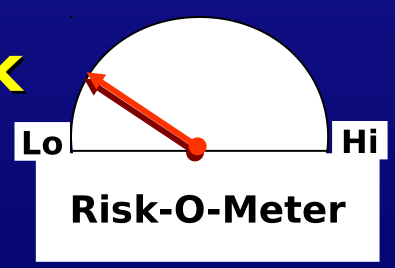


Mitigate Risks (continued)

Defense Acquisition University



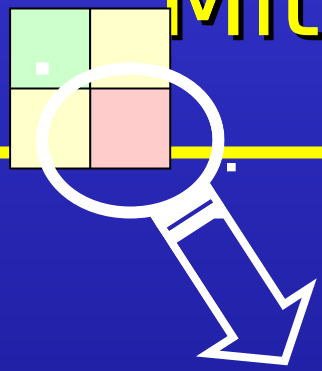
Low Risk

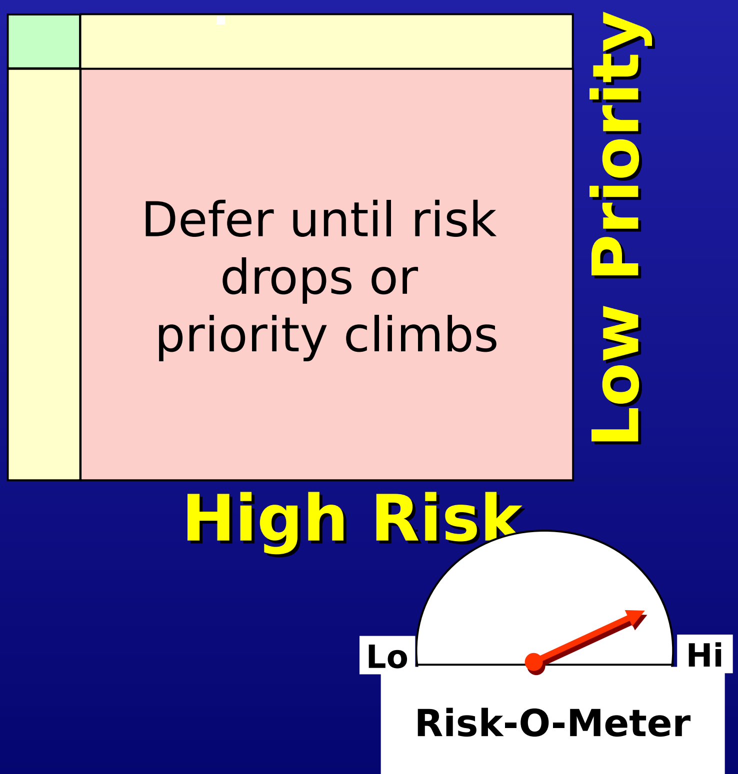


- Don’t commit to production simply based upon low risk
- User Priority should drive production sequence
- Low priority-low risk requirements should be added only after higher priority requirements have been attended to

Mitigate Risks (continued)

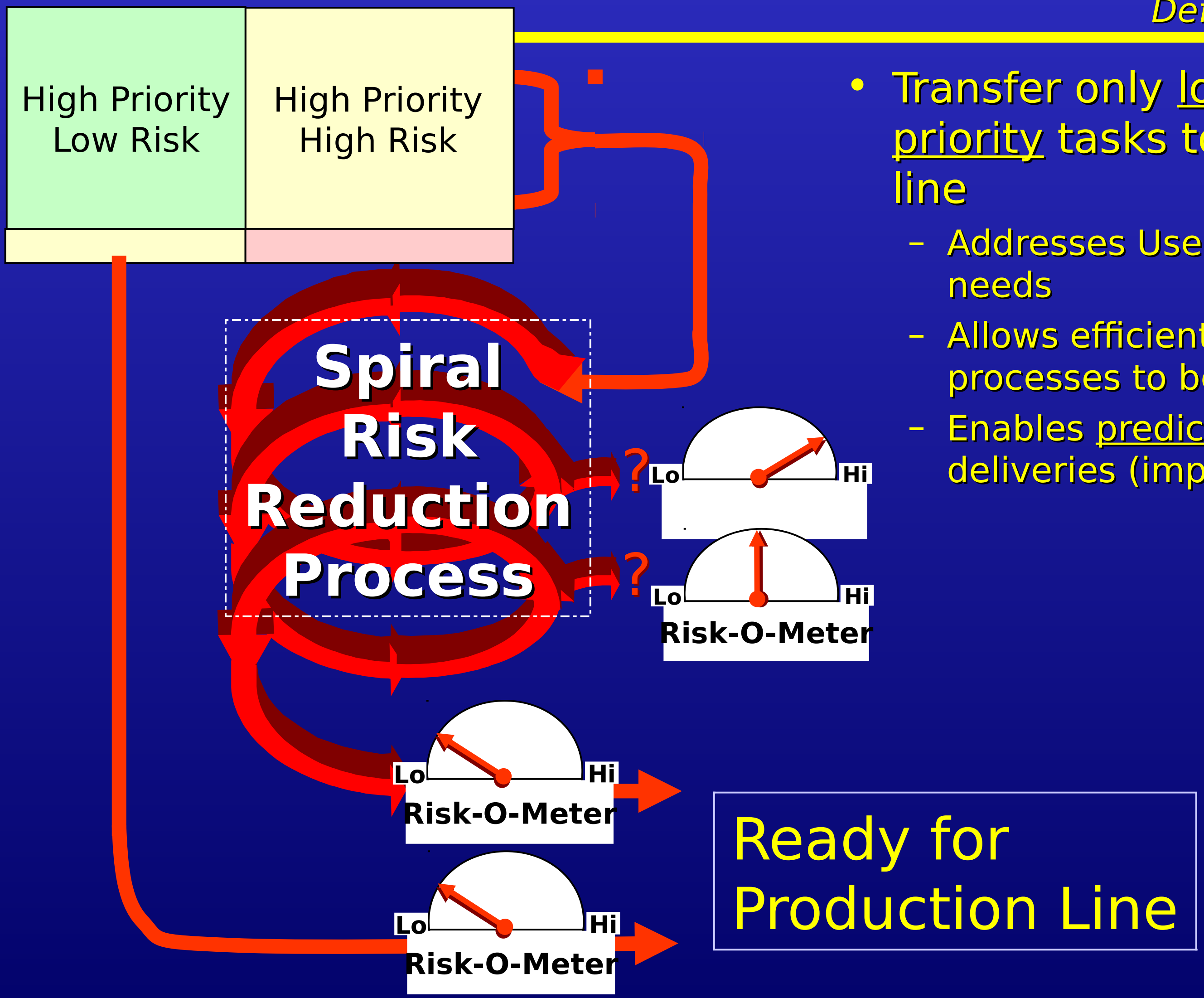
Defense Acquisition University





- Low Priority/High Risk requirements should be deferred, <u>but not discarded</u>
- Priorities and risks are <u>relative</u>, and will change over the implementation period
  - As top priorities are satisfied, the lower priorities will rise
  - As technology progresses, risky requirements will become feasible
  - As new threats emerge, low priority enhancements may become essential
  - As doctrine and policy evolve, so will the OV, giving rise to new, or re-prioritized requirements

Mitigate Risks (concluded)

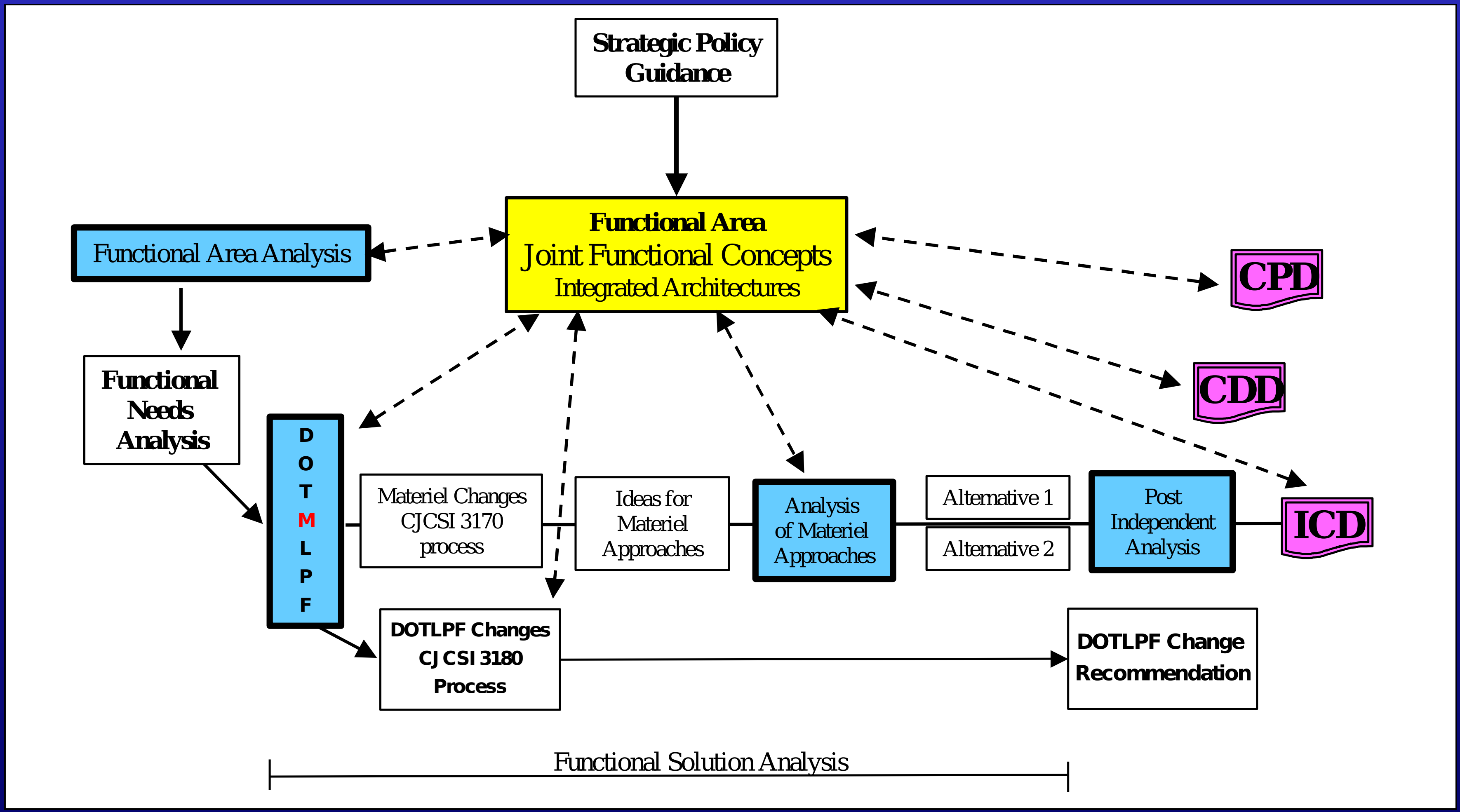


- Transfer only low-risk high-priority tasks to the production line
  - Addresses Users' most pressing needs
  - Allows efficient production processes to be used
  - Enables predictable increment deliveries (important to Users)

Process: Production

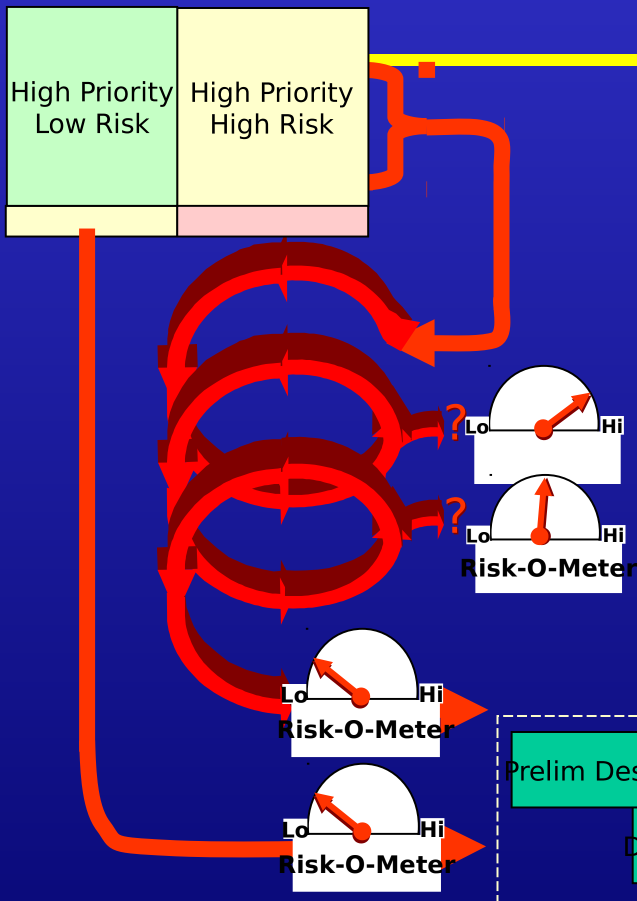
Production: JCIDS Input

Defense Acquisition University

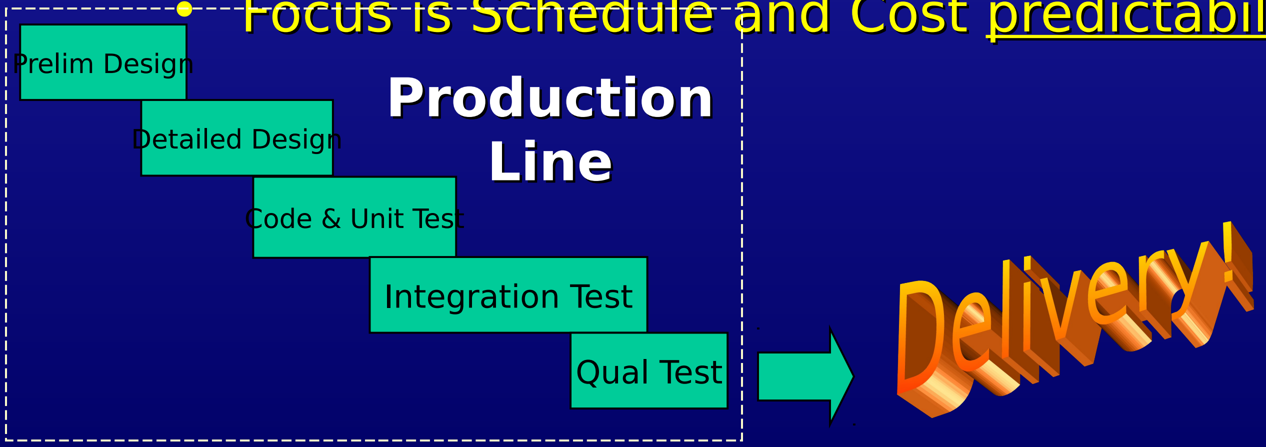


Source: CJCSM 3170.01M FLAG STAFFING DRAFT April 2003

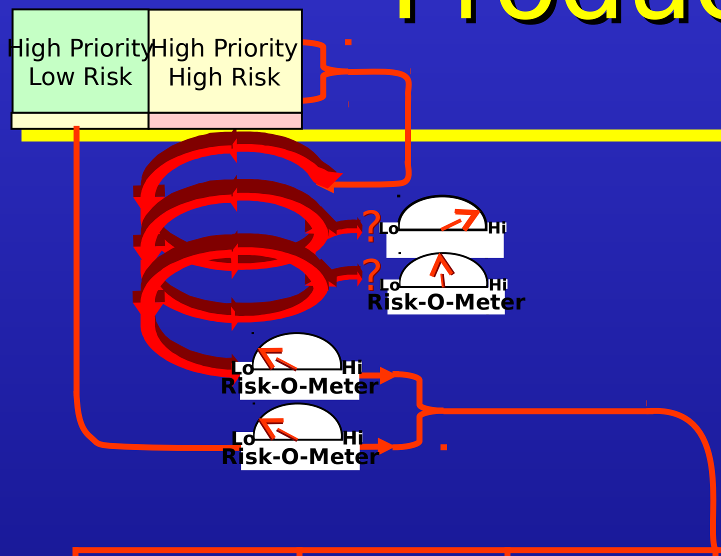
Concept: The Production Line



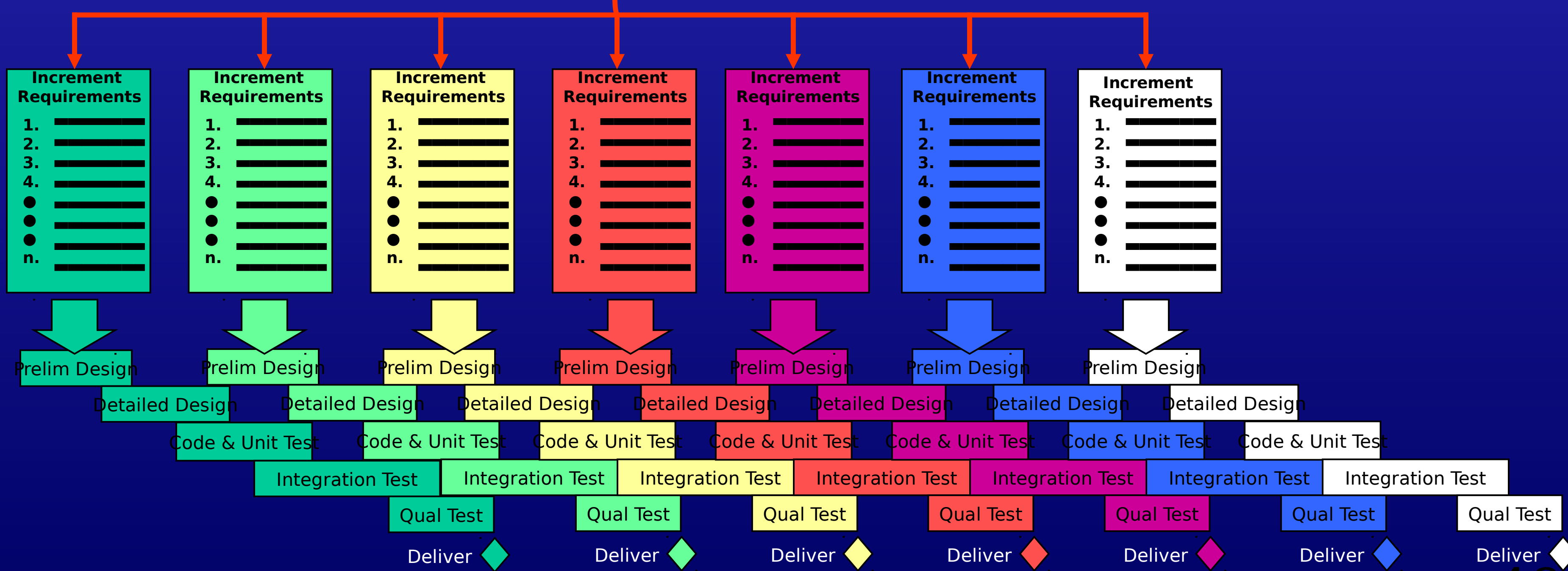
- If risks have been mitigated, highly efficient production processes can be used
  - Tasks have limited, well-defined scope
  - Requirements are stable, low risk to implement
  - Enables application of CM, EVM and other measurement-based management tools
  - Facilitates process maturity, workforce stability
- Focus is Schedule and Cost <u>predictability</u>



Production Process



- Series of incremental builds
- Incorporate high priority low risk requirements
- Efficient development processes
- Reliable product
- Predictable cost and schedule

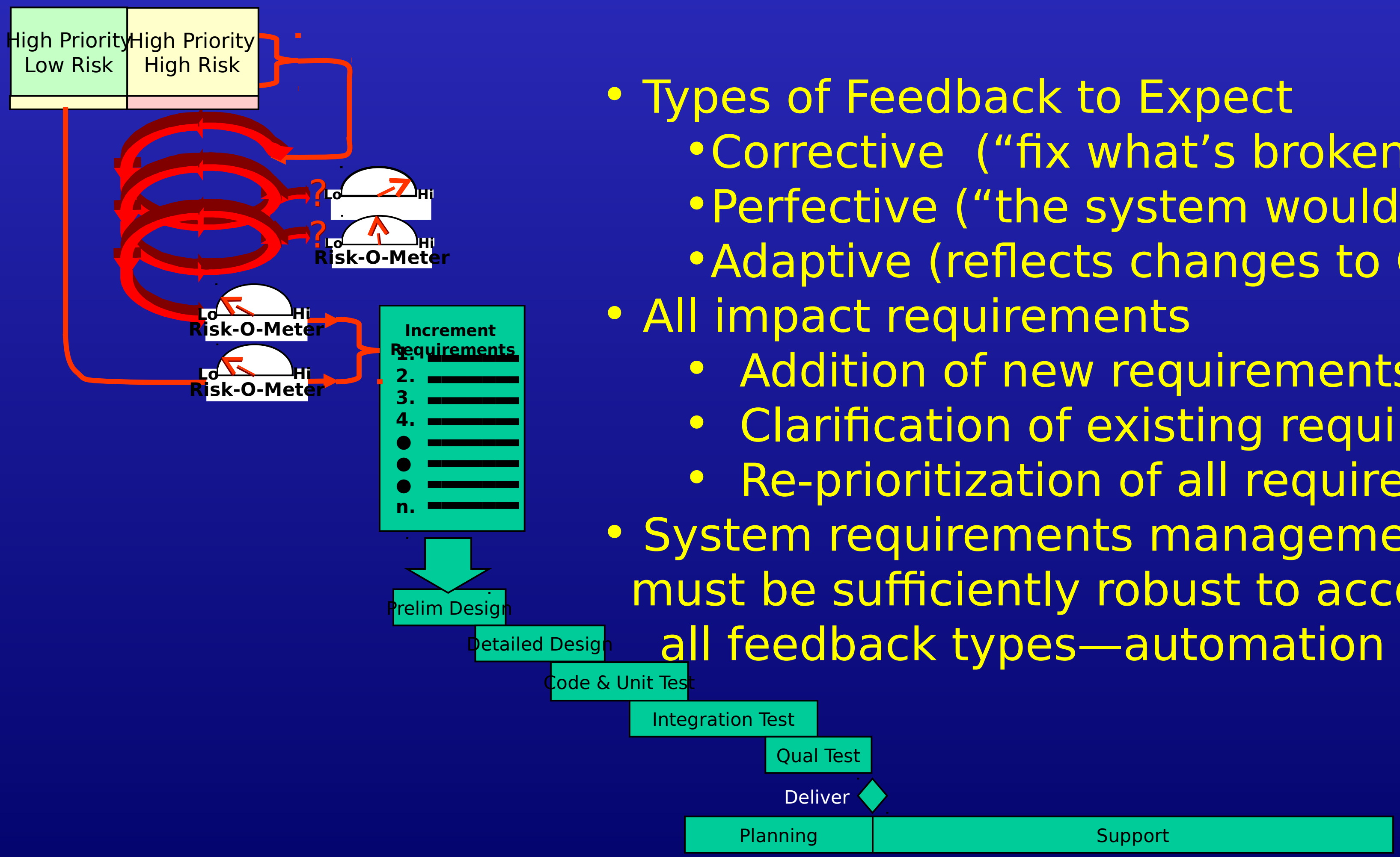


Process: Delivery, Support & Feedback

Key Concepts

- Deliver <u>on-time</u> with promised capability
  - Maintain credibility of acquisition process
- Support what is delivered
- Establish integrated corrective action & requirements management process
  - Distinction between "development" and "support" is not as meaningful in EA
  - Establish regular feedback, requirements development, validation, prioritization meetings
  - Leverage training, testing, experimentation opportunities to gather "ground truth" data

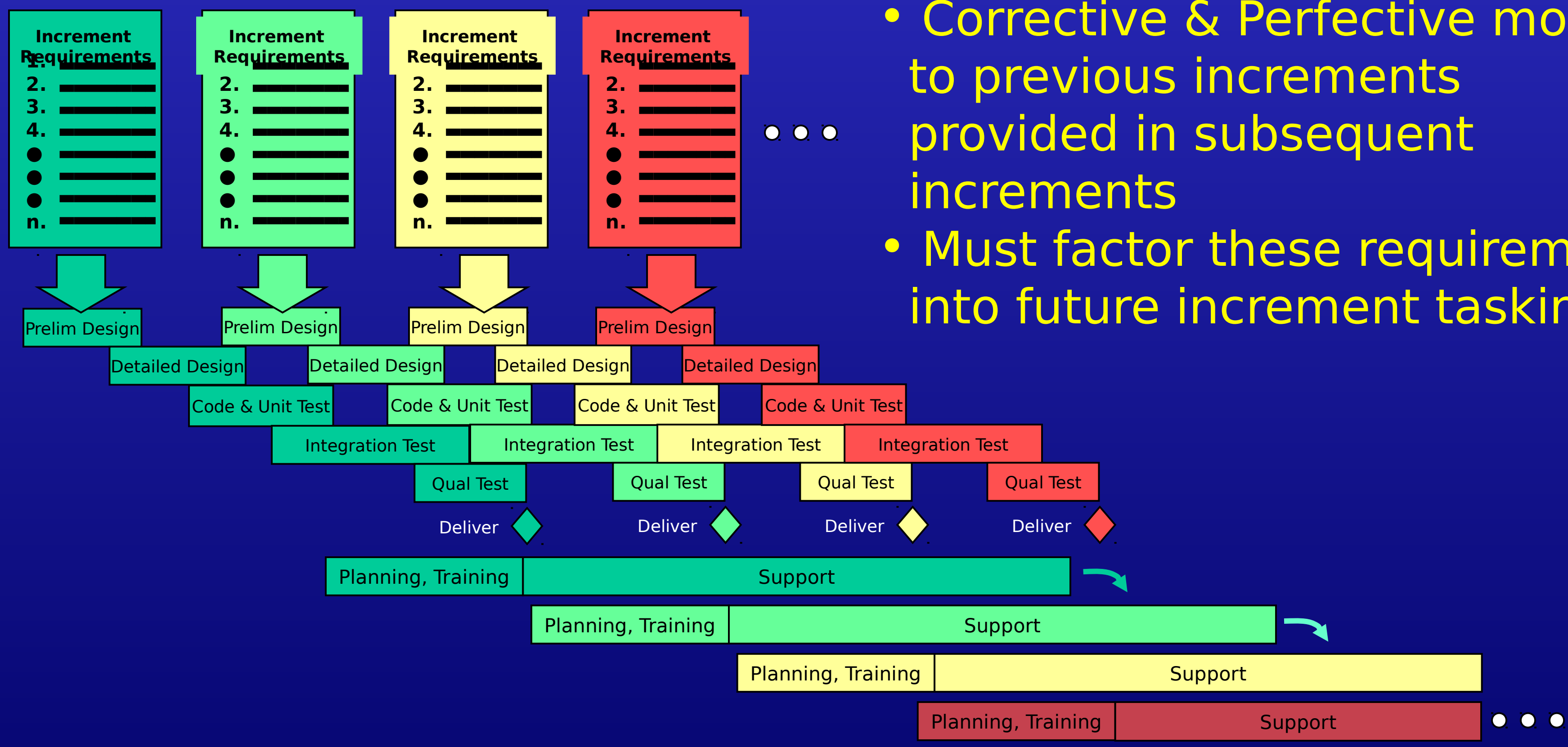
Delivery, Support & Feedback



- Types of Feedback to Expect
  - Corrective ("fix what's broken")
  - Perfective ("the system would be bette
  - Adaptive (reflects changes to OV)
- All impact requirements
  - Addition of new requirements
  - Clarification of existing requirements
  - Re-prioritization of all requirements
- System requirements management proce must be sufficiently robust to accommoda all feedback types—automation is a mus

Supporting Increments

Defense Acquisition University



- Corrective & Perfective mods to previous increments provided in subsequent increments
- Must factor these requirements into future increment tasking

Acquisition Process Summary

- Acquirer must collaborate with user in program-level architecture and requirements definition
- Acquirer must stay engaged with user throughout product lifecycle
- Incorporate business, support, and test planning in the architecture phase
- Establish and distribute program-level SV as first “deliverable”
- Implement functionality in order of operational priority using OV as reference
- Mitigate risk of high-priority requirements using spiral development
- Send only low-risk high-priority requirements to production line
- Integrate test and support early and throughout increment development
- Institute formal support & feedback over lifecycle

Special Interest Items

Contracting Implications

Logistics Implications

Test & Evaluation Implications

Contracting Implications

Contracting For EA

- Source Selection
- Activities
- Contract Types

Source Selection For EA

- Dominant considerations for EA
  - Architecture development/management
    - Ability to develop and support an open, scalable, secure & robust architecture
  - Process management
    - Ability to manage the complex EA process
  - Vendor, subcontractor management
    - Ability to impose interoperability requirements on functional developers
  - Development, integration, & support of functional modules

Contracting For EA

- Activities
  - System program & process management
    - Maintaining contact with stakeholders
    - Maintaining control of the processes and players
  - System architecting, engineering and integration
    - Maintaining control of the architecture
    - Imposing design standards on functional elements
  - Requirements development
  - Risk management & technology transition
  - Development of individual increments
  - Deployment/delivery of individual increments
  - Support and maintenance of delivered increments
  - Incremental & integrated test & evaluation

Contracting Implications of EA

- Some activities are continuous
  - Program and process management
  - System architecting and integration
  - Support and maintenance
- Some activities are discrete
  - Increment development, delivery
- Some activities are episodic
  - Test & Evaluation
- Contract vehicle must be flexible to accommodate all types of activities

Modular Contracting

- Continuous tasks for processes
  - Program and process management
  - System architecting and integration
  - Support and maintenance
- Discrete tasks for deliverables
  - Architecture development and deployment
  - Risk reduction and technology transition
  - Increment development, delivery
- Semi-discrete tasks for episodic activities
  - Test & Evaluation

Continuous Tasks & Processes

- Continuous tasks are processes which span individual increments
  - Program and process management
  - System architecting and integration
  - Support and maintenance
- Emphasis is on continuity and maintenance of system-level expertise
- Criticality of this function depends upon
  - Robustness of Government PMO Team
  - Robustness of System Architecture

Discrete Tasks for Deliverables

- Architecture development and deployment
  - Domain analysis and modeling
  - Analyses of alternatives
  - Feasibility, scalability studies
- Risk reduction and technology transition
  - Prototyping
  - Spiral Development
- Increment development, delivery
  - Pre-negotiated task order labor rates
  - Definitize task scope & deliverables
  - Manage as a discrete project

Episodic Activities

- Require some persistent knowledge from increment to increment
- Levels of activity vary substantially over time
  - Operational Test & Evaluation is a good example
  - Product delivery and deployment may be another
- Maintaining a “standing army” not efficient
- Consider Semi-discrete tasks
  - Pre-negotiated task packages
  - “Ramp-up” provisions to re-allocate staff
  - Provisions to facilitate outsourcing

Contract Types For EA

- Consider CPAF contract for architecture development phase
- Task Order contracts probably the best
  - CPAF tasks for continuous processes
  - T&M tasks for prototyping & risk reduction
  - FPI tasks for production & delivery

Contracting Summary

- Source Selection should be based upon process capabilities more than technical
- Consider architecture as a separate entity
- Modular contracting is essential

Logistics Implications

Logistics Implications of EA

- Time-phased requirements mean that the capabilities of delivered items will evolve over time
- Different capabilities mean different configurations
- Different configurations will make logistics support more complex, and thus more costly
- Look out for interoperability issues!
  - Different units with different versions of the same equipment may not be able to interoperate
  - This may impact support resources

Logistics Role in EA

- To reduce support costs, supportability issues must be addressed early
  - In an IPPD environment
  - To influence architecture and design decisions
    - Open vs. proprietary architecture
    - Modular or tightly-coupled design
  - To influence support concept
    - CLS, PBL, Organic
    - Support multiple CI's or retrofit

Architecting for supportability

- Modular, scalable, open systems architecture minimize the support cost of multiple CI's
  - Core functionality allocated to the architecture
  - Mission-specific functionality allocated to modular elements
  - Functional modules can be changed without extensive impact to architecture or other elements.
  - Support for core will be similar across configurations
  - Unique support requirements limited to functional modules
  - Retrofit/upgrade cost is minimized

Logistics Summary

- Logistics support issues must be addressed early
- Investment in a supportable architecture is vital to minimize support costs
- Critical choices must be made
  - Multiple CI's or retrofit to single CI
  - Organic or Contractor logistics support
  - Can PBL work for your product?

Test & Evaluation Implications

T&E Implications of EA

- Time-phased requirements mean that the capabilities of delivered items will differ from increment to increment
- Each increment must be tested to demonstrate achievement of requirements
- Integrated system must demonstrate operational effectiveness and suitability

T&E's Role in EA

- T&E issues must be addressed early
  - In an IPPD environment
  - To influence architecture and design decisions
    - Open vs. proprietary architecture
    - Modular vs. tightly-coupled design
  - To influence test concept
    - Integrating development and operational test
    - Operational test intervals
    - Degree of regression testing required
  - To influence resource decisions
    - Investment in integrations and test facilities
    - Ensure adequate test resources available

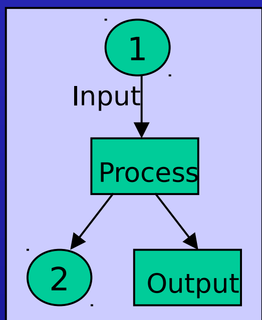
Architecting for Testability

- Modular, open systems architecture may reduce the test impact of EA
  - Core/common functionality allocated to the architecture
  - Mission-specific functionality allocated to modules
  - Functional modules can be added/changed without extensive impact to architecture or other elements.
  - Regression test (re-verifying previously-tested functionality) from increment to increment may be reduced

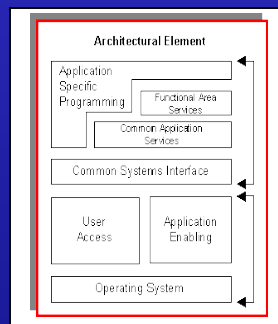
Integrate T&E Considerations Into Requirements Process

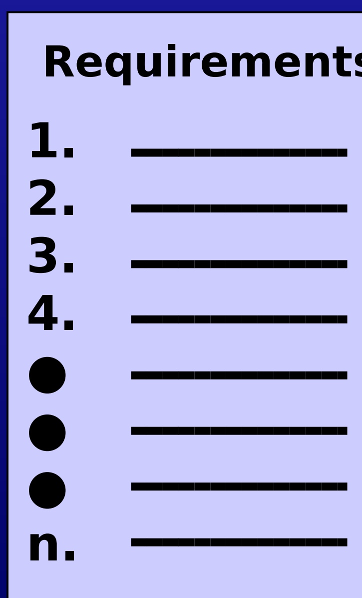
Defense Acquisition University

OV



SV





- Linkage between OV and functional requirements defines operational effectiveness criteria for OT
- This linkage must be maintained as requirements are allocated to modules and increments.
- The criteria and rationale for each requirement feeds the test plan development process

T&E Summary

- T&E issues must be addressed early
- Investment in a modular architecture is vital to minimize T&E costs
- Critical choices must be made
  - Integration of DT and OT
  - OT intervals
  - Amount of regression testing required
  - Program, Test Agency resources allocated to the program